ANALES DEL XIII CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN

WORKSHOPS:
CONGRESO GENERAL
ARQUITECTURAS, REDES Y SISTEMAS OPERATIVOS
INGENIERÍA DE SOFTWARE Y BASE DE DATOS
COMPUTACIÓN GRÁFICA, IMÁGENES Y VISUALIZACIÓN
TECNOLOGÍA INFORMÁTICA APLICADA EN EDUCACIÓN
PROCESAMIENTO DISTRIBUIDO Y PARALELO
AGENTES Y SISTEMAS INTELIGENTES

1 AL 5 DE OCTUBRE DE 2007
ARGENTINA
CHACO Y CORRIENTES

CACIC2007
UTN
UNNE
RedUNCI

Anales del

XIII Congreso Argentino de Ciencias de la Computación (CACIC)

1 al 5 de octubre de 2007 · Corrientes y Resistencia, Argentina

CACIC 2007 Anales del XIII Congreso Argentino de Ciencias de la Computación - 1a ed. -
Corrientes: Universidad Nacional del Nordeste, 2007.
CD-ROM.

ISBN 978-950-656-109-3

1. Computación. 2. Informática.
CDD 005.3

Autoridades

Red de Universidades con Carreras de Informática (Red UNCI)
Coordinadores: Ing. Armando De Giusti, Dr. Guillermo Simari

Universidad Nacional del Nordeste (UNNE)
Rector: Arq. Oscar Valdés
Decano Facultad de Ciencias Exactas: Ing. Eduardo del Valle
Directora Departamento de Informática: Mgter. Gladys Dapozo

Universidad Tecnológica Nacional (UTN)
Rector: Ing. Héctor Brotto
Decano Facultad Regional Resistencia: Ing. Francisco Benítez
Directora Departamento de Informática: Ing. Mirtha Giovannini

Presidente del Comité Científico: Ing. Armando De Giusti
Presidente del Comité Organizador: Mgter. Gladys Dapozo
Directora de la XI Escuela Internacional de Informática: Mgter. Liliana Cuenca Pletsch

Comité Organizador local

Universidad Nacional del Nordeste (UNNE)
Mgter. Gladys Dapozo
Mgter. María del Socorro Foio
Dr. Jorge Emilio Monzón
Mgter. María Viviana Godoy Guglielmone
Mgter. David la Red Martínez
Mgter. Oscar Vallejos
Lic. Ricardo Monzón

Universidad Tecnológica Nacional (UTN)
Mgter. Liliana Cuenca Pletsch
Esp. Mirtha Giovannini
Ing. Carolina Orcola
Ing. Ricardo Calcagno
Srta. Noelia Pinto
Sr. Andrés Arevalo
Srta. Cinthia Quiroga
Sr. Carlos Cuello

Comité Académico

Está conformado por un representante de cada Universidad Miembro de la RedUNCI.

Acosta Nelson
Universidad Nacional del Centro de la Provincia de Buenos Aires

Aguirre Jorge
Universidad Nacional de Río IV

Alfonso Hugo
Universidad Nacional de La Pampa

Arias Figueroa Daniel
Universidad Nacional de Salta

Carmona Fernanda
Universidad Nacional de Chilecito

Ceccato Alejandro
Universidad Nacional de Rosario

Cosso Pablo
Universidad de Buenos Aires
(Facultad de Ingeniería)

Cuenca Pletsch Liliana
Universidad Tecnológica Nacional
(Sede Resistencia – Chaco)

Cukierman Uriel
Universidad Tecnológica Nacional
(Rectorado)

Dapozo Gladys
Universidad Nacional del Nordeste

De Giusti Armando
Universidad Nacional de La Plata

De Vicenzi Marcelo
Universidad Abierta Interamericana

Esquivel Susana
Universidad Nacional de San Luis

Estayno Marcelo
Universidad Nacional de La Matanza

Feierherd Guillermo
Universidad Nacional de la Patagonia S.J. Bosco

Fernández Miguel
Universidad Nacional de Entre Ríos

Forradillas Raimundo
Universidad Nacional de San Juan

Kantor Raúl Eduardo
Universidad Nacional de Rosario

Leone Horacio
Universidad Tecnológica Nacional
(Sede Santa Fe)

Loiseau Irene
Universidad de Buenos Aires
(Facultad de Ciencias Exactas)

Loyarte Horacio
Universidad Nacional del Litoral

Márquez María Eugenia
Universidad Nacional de la Patagonia Austral

Morchio Raul
Universidad Tecnológica Nacional
(Sede Córdoba)

Padovani Hugo
Universidad de Morón

Paganini José Humberto
Universidad Nacional de Jujuy

Parto Laura
Universidad Nacional de Villa María

Peri Jorge
Universidad Nacional de Luján

Sánchez Laura
Universidad Nacional del Comahue

Segura Enrique
Universidad de Buenos Aires
(Facultad de Ciencias Exactas)

Simari Guillermo
Universidad Nacional del Sur

Viales Mario
Universidad Nacional de Misiones

Comité Científico

Abdallah Chaouki (USA)

Acosta Nelson (Argentina)

Aguilar Castro José (Venezuela)

Aguirre Jorge (Argentina)

Alba Torres Enrique (España)

Angel Ed (USA)

Ardenghi Jorge (Argentina)

Baum Gabriel (Argentina)

Bria Oscar (Argentina)

Brisaboa Nieves (España)

Castro Silvia (Argentina)

Cechich Alejandra (Argentina)

Coello Coello Carlos (México)

Chávez Edgar (México)

De Giusti Armando (Argentina)

De Jong Ken (USA)

Díaz Javier (Argentina)

Dosch Walter (Alemania)

Esquivel Susana (Argentina)

Estayno Marcelo (Argentina)

Estévez Elsa (Argentina)

Girardi Rosario (Uruguay)

Gutiérrez Gilberto (Chile)

Hernández Aguirre Arturo (México)

Jordan Ramiro (USA)

Katrib Miguel (Cuba)

Koronacki Jacek (Polonia)

Loiseau Irene (Argentina)

Luque Emilio (España)

Marín Mauricio (Chile)

Margaleff Tomas (España)

Merkle Daniel (Alemania)

Merelo Cuervo J. J. (España)

Mezura Montes Efrén (México)

Michalewicz Zbigniew (Australia)

Middenford Martin (Alemania)

Naiouf Marcelo (Argentina)

Nebro Antonio (España)

Pesado Patricia (Argentina)

Printista Marcela (Argentina)

Randall Gregory (Uruguay)

Ras Zbigniew (USA)

Rito Silva Antonio (Portugal)

Riesco Daniel (Argentina)

Rodríguez León Casiano (España)

Rossi Gustavo (Argentina)

Sánchez Laura (Argentina)

Schoenauer Marc (Francia)

Scolnik Hugo (Argentina)

Señas Perla (Argentina)

Serón Arbeoloa Francisco (España)

Simari Guillermo (Argentina)

Steinmetz Ralf (Alemania)

Tarouco Liane (Brasil)

Tirado Francisco (España)

Zamarro José Miguel (España)

II Workshop de Arquitecturas, Redes y Sistemas Operativos (WARSO)

Coordinadores

Javier Díaz (UNLP)
Antonio Castro Lechtaler (UBA/UTN)

Comité Científico

Acosta Nelson (Argentina)
Bevilacqua Roberto (Argentina)
Castro Lechtaler Antonio (Argentina)
El Saddik Abed (Canada)
Finochietto Jorge (Italia)
Fusario Ruben (Argentina)
Jordan Mario (Argentina)
Marrone Luis (Argentina)
Mercado Gustavo (Argentina)
Obac RodaValentin (Brasil)
Tarouco Liane (Brasil)
Steinmetz Ralf (Alemania)

Temas

Arquitecturas de computadoras
Sistemas embebidos
Arquitecturas orientadas a servicios (SOA)
Redes de comunicaciones
Redes heterogéneas
Redes de Avanzada
Redes inalámbricas
Redes móviles
Redes activas
Administración y monitoreo de redes y servicios
Calidad de Servicio (QoS, SLAs)
Seguridad informática y autenticación, privacidad
Infraestructura para firma digital y certificados digitales
Análisis y detección de vulnerabilidades
Sistemas operativos
Sistemas P2P
Middleware
Infraestructura para grid
Servicios de integración (Web Services o .Net)

II Workshop de Arquitecturas, Redes y Sistemas Operativos (WARSO)

Evaluadores

Acosta Nelson
Universidad Nacional del Centro Prov. de Bs. As. - Argentina

Bevilacqua Roberto
Universidad de Buenos Aires - Argentina

Castro Lechtaler Antonio Ricardo
UTN Regional Bs. As. - Argentina

Díaz Javier
Universidad Nacional de La Plata - Argentina

Echaiz Javier
Universidad Nacional del Sur - Argentina

Finochietto Jorge
Politecnico de Torino - Argentina

Fusario Rubén Jorge
UTN Regional Bs. As. - Argentina

Garcia Rafael Benjamin
Universidad Nacional del Sur - Argentina

Luengo Miguel
Universidad Nacional de La Plata - Argentina

Marrone Luis
Universidad Nacional de La Plata - Argentina

Mercado Gustavo
UTN Regional Mendoza - Argentina

Molinari Lía
Universidad Nacional de La Plata - Argentina

Obac Roda Valentin
Universidade de São Paulo - Brasil

Villagarcia Horacio
Universidad Nacional de La Plata - Argentina

IV Workshop de Ingeniería de Software y Bases de Datos (WISBD)

Coordinadores

Patricia Pesado (UNLP)
Elsa Estévez (UNSur)

Comité científico

Alpuente, Maria (España)
Astudillo, Hernán (Chile)
Cechich, Alejandra (Argentina)
Dosch, Walter (Alemania)
Dujmovic, Jozo (USA)
Estayno, Marcelo (Argentina)
Falappa, Marcelo (Argentina)
Fillottrani, Pablo (Argentina)
George, Chris (United Nations)
Janowski, Tomasz (United Nations)
Riesco, Daniel (Argentina)
Rossi, Gustavo (Argentina)
Sousa Pinto, Jorge (Portugal)
Vizcaino, Aurora (España)

Temas

Ingeniería del Software
Bases de Datos

Evaluadores

Adegboyega Ojo
United Nations University - International Institute for Software Technology

Amaro Silvia N.
Universidad Nacional del Comahue

Aranda Gabriela
Universidad Nacional del Comahue

Banchoff Tzancoff Claudia
Universidad Nacional de La Plata

Bertone Rodolfo
Universidad Nacional de La Plata

Boracchia Marcos
Universidad Nacional de La Plata

Buccella Agustina
Universidad Nacional del Comahue

Capobianco Marcela
Universidad Nacional del Sur

Castro Silvia M.
Universidad Nacional del Sur

Cechich Alejandra
Universidad Nacional del Comahue

Chiotti Omar
Universidad Tecnológica Nacional - Regional Santa Fe

Daniele Marcela
Universidad Nacional de Río Cuarto

Dasso Aristides
Universidad Nacional de San Luis

Diaz Alicia
Universidad Nacional de La Plata

Esponda Silvia
Universidad Nacional de La Plata

Estayno Marcelo
Universidad Nacional de La Matanza

IV Workshop de Ingeniería de Software y Bases de Datos (WISBD)

Evaluadores (continuación)

Estevez Elsa
Universidad Nacional del Sur

Falappa Marcelo A.
Universidad Nacional del Sur

Favre Liliana
Universidad Nacional del Centro Prov.Bs.As.

Feierherd Guillermo
Universidad Nacional de la Patagonia San Juan Bosco

Felice Laura
Universidad Nacional del Centro Prov.Bs.As.

Fillottrani Pablo R.
Universidad Nacional del Sur

Flores Andrés Pablo
Universidad Nacional del Comahue

Funes Ana
Universidad Nacional de San Luis

Gagliardi Olinda Edilma
Universidad Nacional de San Luis

Giandini Roxana S.
Universidad Nacional de La Plata

Gordillo Silvia
Universidad Nacional de La Plata

Herrera Norma Edith
Universidad Nacional de San Luis

Martig Sergio R.
Universidad Nacional del Sur

Martín Adriana
Universidad Nacional del Comahue

Mauco Virginia
Universidad Nacional del Centro Prov.Bs.As.

Mencl Vladimir
Charles University, Republica Checa

Montagna Marcelo
CONICET

Montejano Germán
Universidad Nacional de San Luis

Olsina Luis
Universidad Nacional de La Pampa

Orosco Ricardo
Universidad Argentina de La Empresa

Pesado Patricia
Universidad Nacional de La Plata

Pons Claudia F.
Universidad Nacional de La Plata

Queiruga Claudia
Universidad Nacional de La Plata

Ramón Hugo
Universidad Nacional de La Plata

Reyes Nora
Universidad Nacional de San Luis

Riesco Daniel
Universidad Nacional de San Luis

Rivero Laura
Universidad Nacional del Centro Prov.Bs.As.

Rossi Gustavo
Universidad Nacional de La Plata

Schiavoni Alejandra
Universidad Nacional de La Plata

Silva Leila
Universidade Federal de Pelotas, Brasil

Thomas Pablo
Universidad Nacional de La Plata

Vitturini Mercedes
Universidad Nacional del Sur

V Workshop de Computación Gráfica, Imágenes y Visualización (WCGIV)

Coordinadores

Silvia Castro (UNSur)
Roberto Guerrero (UNSL)

Comité científico

Abásolo María José (España)
Abdallah Chaouki (USA)
Angel, Ed (USA)
Baldassarri, Sandra (España)
Bria, Oscar (Argentina)
Collazos Ordóñez, César Alberto (Colombia)
Escarza, Sebastián (Argentina)
González, María Paula (Argentina)
Jordan, Ramiro (USA)
Larrea, Martín (Argentina)
Latorre, Pedro (España)
Martig, Sergio (Argentina)
Palomar, Manuel (España)
Pina, Alfredo (España)
Randall, Gregory (Uruguay)
Russo, Claudia (Argentina)
Serón Arbeoloa Francisco (España)
Simari, Patricio (Canadá)
Vénere, Marcelo (Argentina)

Temas

Computación Gráfica
Interfaz Hombre-Máquina
Interfaces con Dispositivos No Convencionales
Visualización de Información
Visualización Científica
Visualización de Software
Procesamiento de Señales

V Workshop de Computación Gráfica, Imágenes y Visualización (WCGIV)

Evaluadores

Abásolo María José
Universidad Nacional de La Plata - España

Baldassarri Sandra
Universidad de Zaragoza - España

Bria Oscar
Universidad Nacional de La Plata - Argentina

Castro Silvia
Universidad Nacional del Sur - Argentina

Chesñevar, Carlos
Universidad Nacional del Sur - Argentina

Collazos Ordóñez César Alberto
Universidad del CAUCA - Colombia

Escarza Sebastián
Universidad Nacional del Sur - Argentina

Francisco Serón Arbeloa
Universidad de Zaragoza - España

Gómez, Sergio
Universidad Nacional del Sur - Argentina

González María Paula
Universidad Nacional del Sur - Argentina

Guerrero Roberto
Universidad Nacional de San Luis - Argentina

Larrea Martín
Universidad Nacional del Sur - Argentina

Latorre Pedro
Universidad de Zaragoza - España

Martig, Sergio
Unversidad Nacional del Sur - Argentina

Pina Alfredo
Universidad de Navarra - España

Ponzoni Ignacio
Universidad Nacional del Sur - Argentina

Randall Gregory
Universidad de la República - Uruguay

Russo Claudia
Universidad Nacional de La Plata - Argentina

Simari Patricio D.
Universidad de Toronto - Canadá

Vénere Marcelo
Universidad Nacional del Centro Prov. de Bs. As. - Argentina

VI Workshop de Tecnología Informática Aplicada en Educación (WTIAE)

Coordinadores

Perla Señas
Cristina Madoz

Comité científico

Cabero Julio (España)
De Petris Beatriz (Argentina)
Feierherd Guillermo (Argentina)
Gorga Gladys (Argentina)
Lucero Margarita (Argentina)
Luque Mónica (OEA)
Marcelo Carlos (España)
Maguitman Ana (Argentina)
Malbrán María (Argentina)
Moroni Norma (Argentina)
Otero Rita (Argentina)
de Pablo Pons Juan (España)
Paldao Carlos (OEA)
Pessaq Raúl (Argentina)
Ponzoni Ignacio (Argentina)
Sanz Cecilia (Argentina)
Tartaglia Angelo (Italia)
Vitturini Mercedes (Argentina)
Zamarro José (España)

Temas

Informática Educativa
Tecnologías Informáticas Aplicadas a la Educación

VI Workshop de Tecnología Informática Aplicada en Educación (WTIAE)

Evaluadores

Cataldi Zulma
Universidad de Buenos Aires - Argentina

Chiarani Marcela Cristina
Universidad Nacional de San Luis - Argentina

De Giusti Armando E.
Universidad Nacional de La Plata - Argentina

Depetris Beatriz
UN de la Patagonia San Juan Bosco - Argentina

Feierherd, Guillermo
UN de la Patagonia San Juan Bosco - Argentina

Gorga Gladys
Universidad Nacional de La Plata - Argentina

Lucero María Margarita
Universidad Nacional de San Luis - Argentina

Luque Monica
OEA - Argentina

Madoz Maria Cristina
Universidad Nacional de La Plata - Argentina

Maguitman Ana G.
Universidad Nacional del Sur - Argentina

Moroni Norma
Universidad Nacional del Sur - Argentina

Otero Rita
Universidad Nacional del Centro Prov. de Bs. As. - Argentina

Pessacq Raúl
Universidad Nacional de La Plata - Argentina

Ponzoni Ignacio
Universidad Nacional del Sur - Argentina

Rosanigo Zulema Beatriz
UN de la Patagonia San Juan Bosco - Argentina

Sanchez Laura
Universidad Nacional del Comahue - Argentina

Sanz Cecilia
Universidad Nacional de La Plata - Argentina

Señas Perla
Universidad Nacional del Sur - Argentina

Vittutini, Mercedes
Universidad nacional del Sur - Argentina

Zanconi Marcelo
Universidad Joseph Fourier de Grenoble - Francia

Zangara Alejandra
Universidad Nacional de La Plata - Argentina

VIII Workshop de Procesamiento Distribuido y Paralelo (WPDP)

Coordinadores

Armando De Giusti (UNLP)
Marcela Printista (UNSL)

Comité científico

José Aguilar (Venezuela)
Leandro Bertogna (Argentina)
Héctor Cancela (Uruguay)
Ramón Doallo (España)
Vicente Hernandez (España)
Emilio Luque (España)
Tomás Margalef (España)
Mauricio Marín (Chile)
Marcelo Naiouf (Argentina)
María F. Píccoli (Argentina)
Ana Ripoll (España)
Dolores Rexachs (España)
Casiano Rodríguez León (España)
Josemar Souza (Brasil)
Fernando Tinetti (Argentina)
Francisco Tirado (España)
Juan Touriño (España)

Temas

Sistemas Distribuidos y Paralelos
Algoritmos Paralelos
Computación de altas prestaciones. Supercómputo
Predicción de Performance en Algoritmos Paralelos
Balance de Carga en Sistemas Distribuidos/Paralelos
Escalabilidad
Métricas de Performance y Eficiencia en Paralelismo
Algoritmos sobre Clusters
Computación sobre GRID
Middleware para arquitecturas GRID
Aplicaciones de Sistemas Distribuidos/Paralelos

VIII Workshop de Procesamiento Distribuido y Paralelo (WPDP)

Evaluadores

Aguilar J.L.
CEMISID Universidad de Los Andes - Venezuela

Ardenghi Jorge R.
Universidad Nacional del Sur – Argentina

Bertogna Leandro
Universidad Nacional del Comahue - Argentina

Cancela Héctor
Universidad de la República - Uruguay

De Giusti Armando
Universidad Nacional de La Plata - Argentina

Doallo Ramón
Universidade da Coruña - España

Fabero Juan Carlos
Universidad Complutense de Madrid - España

Echaiz Javier
Universidad Nacional del Sur - Argentina

Hernandez Vicente
Universidad Politécnica de Valencia - España

Garcia Rafael Benjamin
Universidad Nacional del Sur - Argentina

Luque Emilio
Universidad Autónoma de Barcelona - España

Margalef Tomás
Universidad Autónoma de Barcelona - España

Marín Mauricio
Universidad de Magallanes - Chile

Naiouf Marcelo
Universidad Nacional de La Plata - Argentina

Piccoli María Fabiana
Universidad Nacional de San Luis - Argentina

Printista Marcela
Universidad Nacional de San Luis - Argentina

Ramón Hugo
Universidad Nacional de La Plata - Argentina

Rexachs Dolores
Universidad Autónoma de Barcelona - España

Rodrigues de Souza Josemar
Universidade do Estado da Bahia - Brasil

Rodriguez Leon Casiano
Universidad de La Laguna - España

Tinetti Fernando G.
Universidad Nacional de La Plata - Argentina

Tirado Francisco
Universidad Complutense de Madrid - España

Touriño Dominguez Juan
Universidade da Coruña - España

VIII Workshop de Agentes y Sistemas Inteligentes (WASI)

Coordinadores

Guillermo Leguizamón (UNSL)
Marcelo Falappa (UNSur)

Comité científico

Enrique Alba Torres (España)
Carlos A. Coello (México)
Juergen Dix (Alemania)
Susana Esquivel (Argentina)
Lluis Godo (España)
Jacek Koronacki (Polonia)
Juan Julián Merelo (España)
Zbigniew Michalewicz (Australia)
Martín Middendorf (Alemania)
Ronald Prescott Loui (USA)
Iyad Rahwan (Dubay)
Zbigniew Ras (USA)
Carles Sierra (España)
Guillermo Simari (Argentina)
Paolo Rosso (España)

Temas

Metaheurísticas Bio-Inspiradas
Metaheurísticas Basadas en Inteligencia Colectiva
Sistemas Multiagente
Sistemas Inteligentes
Aprendizaje Automático
Inteligencia Artificial
Robótica
Redes Neuronales

VIII Workshop de Agentes y Sistemas Inteligentes (WASI)

Evaluadores

Aguirre Guillermo
Universidad Nacional de San Luis - Argentina

Alba Torres Enrique
Universidad de Málaga - España

Alfonso Hugo
Universidad Nacional de La Pampa - Argentina

Blum Cristian
Universitat Politecnica de Catalunya - España

Brignole Nélida
Universidad Nacional del Sur - Argentina

Cecchi Laura A.
Universidad Nacional del Comahue - Argentina

Chesñevar Carlos Iván
Universidad Nacional del Sur - Argentina

Cobo María Laura
Universidad Nacional del Sur - Argentina

Coello Coello Carlos
Centro de Investigación y de Estudios Avanzados - México

Dix Juergen
Technische Universitaet Clausthal - Alemania

Errecalde Marcelo
Universidad Nacional de San Luis - Argentina

Esquivel Susana
Universidad Nacional de San Luis - Argentina

Falappa Marcelo Alejandro
Universidad Nacional del Sur - Argentina

García Alejandro Javier
Universidad Nacional del Sur - Argentina

Gómez Sergio Alejandro
Universidad Nacional del Sur - Argentina

Hernández-Aguirre Arturo
Centro de Investigaciones Matemáticas - México

Kern-Isberner Gabriele
University of Dortmund - Alemania

Lanzarini Laura
Universidad Nacional de La Plata - Argentina

Leguizamón Guillermo
Universidad Nacional de San Luis - Argentina

Loui Ron P.
Washington University in St. Luis - Estados Unidos

Luque Gabriel
Universidad de Málaga - España

Martinez Diego César
Universidad Nacional del Sur - Argentina

Merelo Juán Julián
Escuela Tec. Sup. de Ingeniería Informática - España

Mezura-Montes Efrén
Laboratorio Nacional de Informática Avanzada - México

Michalewicz Zbigniew
University of North Carolina Charlotte - Estados Unidos

Middendorf Martín
Universidad de Leipzig - Alemania

Nebro Antonio
Universidad de Málaga - España

Pandolfi Daniel
Universidad Nacional de la Patagonia Austral - Argentina

Parra Gerardo A.
Universidad Nacional del Comahue - Argentina

Rahwan Iyad
The British University - Siria, Australia

Roggero Patricia
Universidad Nacional de San Luis - Argentina

Rosso Paolo
Universidad Politécnica de Valencia - España

Simari Guillermo Ricardo
Universidad Nacional del Sur - Argentina

Toscano-Pulido Gregorio
Centro de Investigación y de Estudios Avanzados - México

Vaucheret Claudio A.
Universidad Nacional del Comahue - Argentina

Wagner Neil
Augusta State University - Estados Unidos

Congreso General

Coordinadores

Marcelo Naiouf (UNLP)
Jorge Ardenghi (UNSur)
Aldo Vecchietti (UTN-Sta.Fe)

Temas

Temas de interés en Informática no comprendidos en los Workshops específicos.

Evaluadores

Acosta Nelson
Universidad Nacional del Centro Pcia. de Bs. Aires - Argentina

Aguirre Jorge
Universidad Nacional de Rio Cuarto - Argentina

Ale Mariel
UTN Regional Santa Fe - Argentina

Alfonso Hugo
Universidad Nacional de La Pampa - Argentina

Ardenghi, Jorge
Universidad Nacional del Sur - Argentina

Banchoff Claudia
Universidad Nacional de La Pampa - Argentina

Bria Oscar
Universidad Nacional de La Plata – Argentina

Caliusco María Laura
UTN Regional Santa Fe - Argentina

Cova Walter
Instituto Universitario Aeronáutico - Argentina

Dasso Arístides
Universidad Nacional de San Luis - Argentina

Echaiz, Javier
Universidad Nacional del Sur - Argentina

Fillotrani, Pablo
Universidad Nacional del Sur - Argentina

Galeotti Juan Pablo
Universidad Nacional de Buenos Aires - Argentina

Galli María Rosa
UTN Regional Santa Fe - Argentina

García, Rafael Benjamín
Universidad Nacional del Sur - Argentina

Giacomantone Javier
Universidad Nacional de La Plata - Argentina

Golobisky Maria Fernanda
UTN Regional Santa Fe - Argentina

Gonnet Silvio
UTN Regional Santa Fe - Argentina

Lanzarini Laura
Universidad Nacional de La Plata - Argentina

Leguizamón Guillermo
Universidad Nacional de San Luis - Argentina

Marcela Vegetti
UTN Regional Santa Fe - Argentina

Martinez Diego
Universidad Nacional del Sur - Argentina

Padovani Hugo René
Universidad de Morón - Argentina

Pons Claudia
Universidad Nacional de La Plata - Argentina

Rodríguez María Analía
UTN Regional Santa Fe - Argentina

Roldán Luciana
UTN Regional Santa Fe - Argentina

Señas, Iris Perla
Universidad Nacional del Sur - Argentina

Simari Guillermo
Universidad Nacional del Sur - Argentina

Stegmayer Georgina
UTN Regional Santa Fe - Argentina

Vechietti Aldo
UTN Regional Santa Fe - Argentina

Vegetti Marcela
UTN Regional Santa Fe - Argentina

Venere Marcelo
Universidad Nacional del Centro Prov. de Bs. As. - Argentina

Villagarcía Wanza Horacio
Universidad Nacional de La Plata - Argentina

Villareal Pablo
UTN Regional Santa Fe - Argentina

Índice de artículos

II Workshop de Arquitecturas, Redes y Sistemas Operativos

IV Workshop de Ingeniería de Software y Bases de Datos

V Workshop de Computación Gráfica, Imágenes Y Visualización

VI Workshop de Tecnología Informática Aplicada en Educación

VIII Workshop de Procesamiento Distribuido y Paralelo

VIII Workshop de Agentes y Sistemas Inteligentes

Congreso General

Algoritmo de encaminamiento para redes de sensores inalámbricas utilizando técnicas de agregación

Rosa M. Corti[*]
e-mail: rcorti@fceia.unr.edu.ar

Estela D´Agostino[*]
e-mail: estelad@fceia.unr.edu.ar

Enrique E. Giandomenico[*]
e-mail: giandome@fceia.unr.edu.ar

Roberto M. Martinez[*]
e-mail: rmartinez@dsi.fceia.unr.edu.ar

Abstract

Different environments supervision and control needs local variables measurements. When the places where these measurements are going to be made are of hard access it is used to use Wireless Sensor Networks (WSN). A hierarchical routing algorithm in two levels is proposed in this work to periodically collect the environment's data. Clusters which headers are rotated to uniform the power consumption are defined in the first level. The environment is divided in a grid which is used to locate the headers and to apply aggregation techniques. An intercluster communication structure is defined in the second level. Two well defined rounds are used to send the data to the base station, one to define the enrouting tree and the other to send the collected data. Finally, two kinds of critical network failures are considered: fail in the header election and fail in the inter header communication structure. In both cases solutions are proposed to mitigate the negative effects on the system performance.

Keywords: Comunication Networks, Wireless Comunication, Sensor Networks, Routing, Aggregation.

Resumen

La supervisión y control de entornos de distinta índole implica la necesidad de medir variables propias del ambiente. Cuando los sitios en que deben hacerse estas mediciones son de difícil acceso se suele recurrir a redes inalámbricas de sensores inteligentes (RISI). En este trabajo se propone un algoritmo de encaminamiento jerárquico en dos niveles que realiza una recolección periódica de datos del ambiente. En el primer nivel se definen clusters cuyas cabeceras se rotan para lograr un consumo de energía más uniforme. Se divide la red en una cuadrícula que se utiliza para ubicar las cabeceras y aplicar técnicas de agregación. En el segundo nivel se define la estructura de comunicación intercluster. Para el envío de los datos hacia la estación base se trabaja en dos rondas bien definidas, una de establecimiento del árbol de encaminamiento y otra de envío de la información recolectada. Finalmente, se consideran dos tipos de fallos críticos para la red: fallo en la elección de cabeceras y fallo en la estructura de comunicación entre cabeceras. En ambos casos se proponen soluciones para mitigar los efectos adversos en el funcionamiento del sistema.

Palabras claves: Redes de comunicación, Comunicación inalámbrica, Redes de sensores, Algoritmos de encaminamiento, Agregación de datos.

* Facultad de Cs. Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Rosario, Santa Fé (2000), Argentina.

1 INTRODUCCIÓN

La medición de variables del medio es fundamental para la supervisión y control de entornos y actividades de la más diversa índole. Se realiza en aplicaciones industriales, médicas, agrícolas, de preservación del medio natural o creación de ambientes inteligentes, entre otras. En muchas de las aplicaciones mencionadas, el sensado de las variables de interés debe realizarse en ambientes distantes u hostiles que hacen muy difícil el cableado y la atención periódica de los dispositivos de medición. Por estos motivos se han comenzado a utilizar las redes inalámbricas de sensores inteligentes (RISI) para obtener los datos necesarios.

Las RISI son un tipo particular de redes Ad-Hoc, constituidas por nodos que además de recolectar datos del ambiente, son capaces de procesarlos y colaborar con sus vecinos para transmitirlos hacia la/las estaciones base. Estas redes se auto-organizan para adaptarse a topologías cambiantes, y deben trabajar bajo fuertes restricciones de energía, tratando de maximizar su tiempo de vida útil [1] [2].

Una RISI realiza dos actividades fundamentales para llevar hasta el nodo base información relevante para la aplicación. La primera se ocupa del sensado y procesamiento de la información, y la segunda de la diseminación de la misma en la red. Ambas tareas consumen energía, que como ya se mencionó es un recurso que debe ser cuidadosamente preservado. Se ha establecido que la transmisión consume la mayor parte de la energía disponible, así que se intenta minimizar la tarea de diseminación realizando todo el procesamiento local posible [3]. En este sentido se han desarrollado técnicas de agregación de datos que permiten que el procesamiento de la información se realice en forma distribuida en nodos de la red. La eficiencia y aplicabilidad de estas técnicas, dependen del problema a abordar, por lo que deben ser cuidadosamente elegidas y posiblemente adaptadas teniendo siempre en cuenta el dominio de trabajo [4].

En este trabajo presentamos un algoritmo de encaminamiento para redes de sensores basado en clusters, que realiza una recolección periódica de datos del ambiente. El entorno a supervisar se divide en una cuadrícula uniforme para ubicar las cabeceras de clusters. El algoritmo trabaja en dos rondas bien definidas, una de selección de cabeceras y formación de clusters, y otra de envío de información hacia la estación base utilizando alguna función de agregación. Este artículo se organiza de la siguiente forma: en la sección 2 se analizan las técnicas de agregación de datos, la sección 3 trata sobre el control de topología centrándose en el uso de clusters. En la sección 4 se describe el algoritmo desarrollado y finalmente en las secciones 5 y 6 se enumeran las conclusiones alcanzadas y se proponen líneas de trabajo futuro.

2 AGREGACIÓN DE DATOS

Las RISI están constituidas por una gran cantidad de nodos, y por lo tanto es impracticable utilizar algoritmos de encaminamiento centrados en direcciones, en los cuales es necesario identificar cada nodo unívocamente [3]. Por este motivo, los algoritmos que se utilizan habitualmente en redes de sensores inalámbricas son centrados en datos. En los protocolos

centrados en datos, generalmente se busca aplicar algún tipo de procesamiento en los nodos de forma de reducir el tiempo de transmisión [5].

En este sentido se han desarrollado técnicas de agregación de datos, con el fin de lograr el mensaje más compacto posible que represente al conjunto de mensajes individuales. El mensaje agregado se transmite a través de la red en lugar del conjunto de mensajes procesados [5].

Al incorporar agregación, los algoritmos prolongan la vida útil de la red, reduciendo la cantidad de mensajes a difundir. La reducción del consumo de energía depende en gran medida de la aplicación a encarar, pues son sus características las que determinan la función de agregación a utilizar. Algunas aplicaciones admiten agregados sencillos como la obtención del valor máximo, mínimo o promedio. El estudio de otros fenómenos en cambio, requiere de la transmisión de todos los valores obtenidos o gran parte de ellos. Estos últimos casos son los más desafiantes al momento de diseñar alguna técnica de agregación, que manteniendo los errores acotados, permita cumplir con los requerimientos de la aplicación [6].

Si bien el ahorro de energía es un requerimiento fundamental para el diseño de una técnica de agregación, los algoritmos que la utilizan son evaluados también en otros aspectos [7]:

- Vigencia de los datos: diferencia entre el momento de obtención del dato y el de llegada a la estación base.
- Proporción de lecturas recibidas: Cantidad de lecturas recibidas por la estación base respecto del número total de lecturas realizadas.

Desde ya, el peso que se le dé a cada requerimiento dependerá de la aplicación particular. El problema a abordar será el que determine la magnitud de error aceptable, el período admisible de espera para la llegada de los datos o el porcentaje de mensajes que pueden perderse, sin afectar gravemente la eficiencia esperada para la aplicación.

2.1 Clasificación de las técnicas de agregación

Este trabajo se focaliza en aplicaciones donde es habitual realizar una adquisición periódica de datos, con el fin de mantener un estado de situación actualizado del fenómeno a sensar. Se habla de “rondas de recolección o sensado”, donde los nodos obtienen los valores de las variables de interés y los transmiten hacia la estación base. Estas “rondas” se repiten en períodos cuya duración queda determinada por la naturaleza del fenómeno bajo estudio. En general, podemos clasificar las técnicas de agregación periódicas como [4] [7]:

- Periódica simple: En este tipo de algoritmos, cada nodo espera un período fijo, agrega los datos recibidos en un único paquete de datos y transmite. Si algún nodo se retrasa en el envío de datos o falla, su información no formará parte del agregado enviado hacia la estación base. Un ejemplo muy conocido es el algoritmo de difusión dirigida [8].
- Periódica por salto: En estos algoritmos, cada nodo sabe quienes de entre sus vecinos le

enviarán información (nodos hijos). Entonces, los nodos agregan los datos recibidos y los envían cuando han colectado la información de todos sus hijos. En caso de que algún hijo no se reporte, se espera un período predeterminado, y si no se obtiene respuesta, se difunde el agregado logrado hasta ese momento. El algoritmo que proponemos en este trabajo pertenece a esta categoría.

• Periódica por salto ajustado: Estos algoritmos utilizan el mismo principio que los de agregación periódica por salto, pero incorporan un esquema de retardos para la difusión de los mensajes, basado en la posición de cada nodo en el árbol de distribución. El algoritmo propuesto por Solís [7] cae dentro de esta categoría.

3 CONTROL DE TOPOLOGÍA DE LA RED BASADO EN CLUSTERS

Las RISI son redes densamente pobladas donde cada nodo posee un número muy elevado de potenciales vecinos. Además, es habitual que la red deba reconfigurarse ante pequeñas modificaciones en su topología. Estas características ocasionan problemas para el funcionamiento de la red: se eleva la probabilidad de interferencias y el costo de reconfiguración es alto por la existencia de un gran número de rutas posibles. Una solución para estos inconvenientes consiste en limitar el número de vecinos, lo que se conoce como control de topología. Una de las opciones para controlar la topología es definir redes jerárquicas basadas en clusters o grupos locales de nodos. Para el funcionamiento de los clusters se suele seleccionar nodos con roles especiales conocidos como cabecera (CH). Los CH organizan la comunicación de los nodos miembro del cluster.

La utilización de clusters suma ventajas al control de topología. Por un lado se gestionan los recursos en forma local lo que facilita la escalabilidad de la red. Además estos grupos de nodos están orientados al uso de la técnica de agregación con el fin de reducir el consumo[5]. La ubicación de los puntos de agregación tiene una gran influencia sobre el funcionamiento de los algoritmos que utilizan esta técnica. [9].

3.1 El Protocolo LEACH

El protocolo desarrollado por Heinzelman [10] conocido con el nombre de “Low Energy Adaptive Clustering Hierarchy” (LEACH) propone la utilización de clusters e introduce la rotación del rol de cabecera entre los nodos de la red. Selecciona en forma random algunos nodos para cumplir el rol de CH. Éstos son los encargados de aplicar alguna función de agregación para obtener un único mensaje a enviar al sink. Esto le permite lograr una distribución más uniforme del consumo de energía prolongando el tiempo de vida del sistema. La recolección de datos se realiza en forma periódica y por lo tanto es un protocolo que se comporta en forma correcta cuando se necesita una supervisión constante en un entorno dado [8].

Su funcionamiento se divide en dos fases, una de establecimiento del árbol de encaminamiento y otra de transmisión de datos hacia la estación base. En la primera fase se elige, mediante un algoritmo distribuido que utiliza probabilidades, las cabeceras de cluster. Éstas notifican que asumieron el rol de cabecera y cada nodo se incorpora al cluster cuya señal recibe con más fuerza.

En la segunda fase los nodos miembros le envían sus datos a la cabecera y ésta aplica alguna función de agregación y transmite el mensaje resultante directamente a la estación base. Se asume que todos los nodos miembro están a un salto de la cabecera. Por lo tanto ésta realiza una planificación de tiempos y cada miembro transmite en el periodo de tiempo que le corresponde para evitar colisiones.

4 ALGORITMO PROPUESTO

El algoritmo que se propone utiliza la idea de rotación de cabeceras de cluster propuesta por Heinzelman para el algoritmo descrito en la sección 3.1. Sin embargo, un inconveniente de LEACH es que al seleccionar las cabeceras en forma random, las mismas pueden quedar concentradas en algún sector de la red. Esto podría ocasionar la desconexión de un número indeterminado de nodos, como se puntualiza en [8]. Para subsanar este problema, basándose en las características de las aplicaciones de interés para este trabajo, se propone la división del entorno a supervisar en una cuadrícula con divisiones uniformes y adyacentes como se muestra en la figura 1. Las ventajas de este enfoque se detallarán en las subsecciones siguientes.

4.1 Generalidades del algoritmo

Este algoritmo se desarrolló para aplicaciones de supervisión ambiental con recolección periódica de datos. En este dominio las variables poseen una fuerte correlación espacio/temporal de valores entre nodos vecinos. Para aprovechar esta característica se propone utilizar clusters aplicando alguna función de agregación que dependerá de la aplicación particular. Otras características de la red sobre la que trabajará el algoritmo son:

- Nodos fijos y homogéneos; con el mismo nivel de energía inicial.
- Densidad de nodos uniforme.
- Nodo sink único fuera del área monitoreada.

La ubicación de cada nodo se conoce inicialmente y por lo tanto es posible dividir la zona bajo estudio en una cuadrícula uniforme, identificando cada nodo dentro de las divisiones de la misma. Este enfoque persigue dos objetivos: asegurar que no queden nodos desconectados para una disposición dada de CH, y mantener acotada la cantidad de saltos necesarios para que los datos de los nodos miembro alcancen la CH correspondiente.

El uso de una cuadrícula virtual ya fue propuesto en [9] por Al-Karaki. En su algoritmo, los clusters coinciden con las divisiones de la cuadrícula, y el rol de CH se rota entre los nodos miembros de la misma teniendo en cuenta su energía remanente. En el algoritmo que aquí se propone, cuando se rotan las CH, se vuelven a definir los clusters, pudiendo incorporarse nodos de otras divisiones de la cuadrícula. Esto involucra un mayor número de mensajes de control, pero dichos mensajes son muy breves. Lo que se busca minimizar al introducir clusters dinámicos es el número de transmisiones de datos, al disminuir la cantidad de saltos necesarios. Los mensajes con información del ambiente son mucho más extensos que los de control, y por este motivo se considera beneficioso reducir el tráfico de los primeros a cambio de un incremento de los últimos.

Por último, el funcionamiento del algoritmo se divide en dos rondas bien diferenciadas. La primera se ocupa del establecimiento del árbol de encaminamiento (R1) y la segunda del envío de datos al sink (R2). Estas rondas se repiten periódicamente durante todo el tiempo de vida de la red.

4.2 Establecimiento del árbol de encaminamiento

En cada división de la cuadrícula se seleccionará una cabecera de cluster, por lo que existirán tantos clusters como divisiones en la cuadrícula como puede verse en la figura 1.

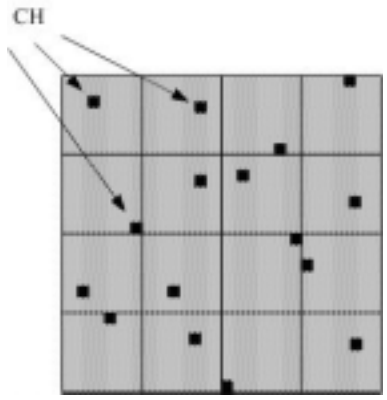


Figura 1: cuadrícula genérica

La ronda R1 del algoritmo se ocupa de definir la estructura de los clusters y la comunicación intra e intercluster. Para ello se encaran diversas tareas que se describen a continuación.

4.2.1 Selección de cabeceras

Los nodos de cada división están identificados con un ID. Cuando comienza a trabajar la red, el nodo con mayor ID de cada partición de la cuadrícula elige en forma random el primer cabecera. En las rondas siguientes cada cabecera elige en forma random a su sucesor y lo notifica. Cada nodo de la red conoce el mayor ID de su división y al asumir el rol de cabecera se autoelimina de una lista de candidatos a CH que recibe de su predecesor. Cuando todos los nodos de una división han cumplido con el rol de cabecera, todos serán nuevamente posibles candidatos. El cabecera que al autoeliminarse de la lista de candidatos obtiene una lista vacía, le advierte al nodo de mayor ID de su división este hecho y se repite el proceso.

4.2.2 Formación de clusters

El árbol de encaminamiento jerárquico se definirá en dos niveles. El primer nivel establecerá la estructura de cada cluster. Luego de seleccionar las cabeceras, cada nodo miembro decidirá a que cluster pertenece. Para ello cada CH debe enviar un mensaje de estructura de cluster (EC) como se muestra en la figura 2.

Tipo mensaje	ID cabecera	Emisor	Nivel Emisor

Figura 2: Estructura del mensaje tipo EC

En este contexto es de gran importancia el concepto de nivel de cada nodo, que es la menor cantidad de saltos que lo separan de su cabecera. De la misma forma resulta relevante que cada nodo identifique al vecino que usará para enviarle mensajes a su CH, al que se llama Ruta Al Sink (RAS).

Los nodos no cabecera que escuchen los mensajes EC se incorporarán al cluster del primer CH que escuchen. Luego reenviarán el mensaje EC colocando su ID como emisor y anunciando su nivel, que será el del emisor mas uno. Los nodos que no escuchan ninguna cabecera eligen como RAS a un nodo común, es decir dentro de un cluster la comunicación será multihop. Un nodo puede recibir mensajes EC con diferentes emisores. Puede tomar entonces dos actitudes, desechar el mensaje si su nivel no mejora, o actualizar su nivel y reenviar el mensaje. Para que un nodo mejore su nivel se debe cumplir que el nivel del nuevo nodo emisor es menor que el nivel actual del nodo menos uno.

Las CH deben conocer a sus hijos, por lo que debe existir un mensaje de pertenencia al cluster que envía cada miembro del mismo. Este mensaje se enviará cuando se ha cumplido un tiempo de espera sin escuchar mensajes de tipo EC. Una vez enviado el mensaje de pertenencia, el nodo no puede cambiar de cluster.

La rotación de cabeceras tiende a lograr una distribución uniforme del gasto de energía. Podría ocurrir que los clusters tengan distinta cantidad de miembros. Es importante destacar, que un nodo común que está en una división de la cuadrícula puede sumarse a un cluster cuya cabecera pertenece a otra. En este caso un cluster quedará formado por nodos que pertenecen a más de una división como se muestra en la figura 3.

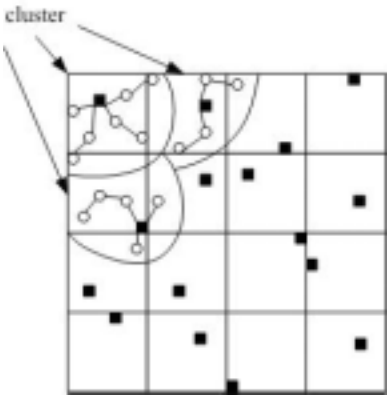


Figura 3: Ejemplo de formación de cluster

4.2.3 Comunicación entre cabeceras

El segundo nivel del árbol de encaminamiento definirá la ruta de comunicación al sink de

todas las cabeceras de cluster. Se establece con el objetivo de que la información agregada alcance al nodo sink, ya que el área a supervisar es de tamaño variable y se puede dar el caso de que no todas las cabeceras alcancen la estación base. En este caso es necesario que las cabeceras lleguen a la estación base a través de su comunicación con otras cabeceras, lo que significa que la comunicación entre las cabeceras y el sink será multihop.

Para establecer la estructura de comunicación entre los CH, el sink envía un mensaje por inundación que se identificó como mensaje tipo E. Este mensaje sólo será tenido en cuenta por los nodos cabecera de cluster, y tendrá la estructura mostrada en la figura 4.

Figura 4: Estructura del mensaje tipo E

Cada nodo cabecera que recibe un mensaje tipo E, adopta como RAS al nodo que figura en el tramo Emisor, y su nivel lo calcula sumándole uno al nivel del Emisor. Luego reenvía el mensaje, colocándose él como nodo emisor [11].

Un nodo puede recibir mensajes de este tipo más de una vez desde diferentes nodos cabecera. Entonces su comportamiento es análogo al descrito en el primer nivel de la jerarquía: desecha el mensaje si su nivel no mejora, o actualiza su nivel y lo notifica.

Por lo tanto, luego de la circulación de este mensaje entre los distintos nodos cabecera, queda perfectamente definida la estructura de comunicación de los CH. Al finalizar la definición de los dos niveles del árbol de encaminamiento, se procede al envío de datos hacia la estación base.

4.3 Envío de datos al sink

Los datos son enviados en forma periódica durante la ronda R2 por los miembros del cluster a la cabecera. Esta lleva a cabo la función de agregación y envía el mensaje resultante a su RAS. De esta forma los datos llegarán a la estación base. El algoritmo desarrollado de acuerdo a la clasificación propuesta por Solis en [7] es periódico por salto. El mensaje de datos intracluster tendrá una estructura como se muestra en la figura 5.

Figura 5: Estructura del mensaje de datos intracluster

La comunicación intercluster puede realizarse teniendo en cuenta dos enfoques distintos. Uno es que las cabeceras al recibir mensajes ya agregados de otras cabeceras los reenvíen directamente hacia el sink y el otro es que vuelvan a aplicar una función de agregación y luego envíen el mensaje agregado. En principio, el primer enfoque implica mayor consumo

de energía ya que se transmiten mayor cantidad de mensajes. El enfoque que se adopte depende fuertemente de la función de agregación a realizar sobre los datos, la cual a su vez será definida de acuerdo a las necesidades de la aplicación. Por lo tanto, la estructura del mensaje de datos intercluster dependerá del enfoque adoptado.

4.4 Potencia de transmisión

La potencia de transmisión de los nodos será variable dependiendo del tipo de comunicación a establecer. Para la comunicación intercluster los nodos cabecera transmitirán con mayor potencia, para lograr la comunicación entre ellos y el sink. El intercambio de mensajes intracluster se podrá realizar con una potencia de transmisión menor. Se fija un radio de transmisión r y las divisiones de la cuadrícula tendrán una longitud 2r como se muestra en la figura 6. De aquí se concluye que:

$$d = \sqrt{(2r)^2 + (2r)^2}$$
$$d = 2\,r\sqrt{2} \quad < 3r$$

Por lo tanto en la comunicación intracluster la mayor distancia entre un CH y un nodo será menor a 3 saltos.

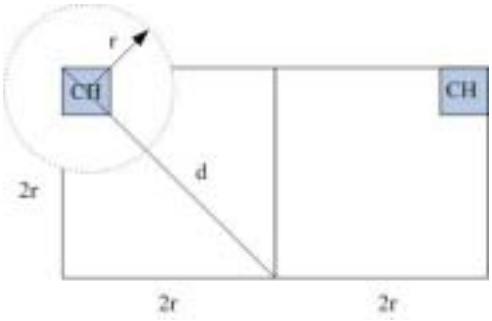


Figura 6: Radio de transmisión intracluster

4.5 Fallos en la red

Durante cada una de las tareas que se desarrollan en las dos rondas del funcionamiento del algoritmo, pueden presentarse fallos. El protocolo desarrollado no da seguridad respecto de la llegada de mensajes a la estación base. Este criterio se adoptó pues para el tipo de aplicaciones de interés es suficiente con fijar un umbral, que representa un porcentaje de pérdida de mediciones admisible, que una vez superado indica que el tiempo de vida de la red ha expirado. Sin embargo, en la ronda R1 de establecimiento del árbol de encaminamiento existen ciertos fallos, que de producirse degradan en forma significativa el funcionamiento de la red. Se considerarán como fallos críticos los que se producen en las etapas de elección de cabeceras y definición de la estructura de comunicación de cabeceras. Para disminuir el impacto de estos fallos se propone los siguientes cursos de acción.

- **fallos en la elección de cabeceras:** la cabecera actual de la cuadrícula elige en forma random a su sucesora y le envía la lista de candidatos a CH. El nodo elegido responde con un mensaje de aceptación. La cabecera actual espera por el mensaje de aceptación un Δt, en caso de no recibirlo elige una cabecera sustituta.

- **fallos en la definición de la estructura de comunicación de cabeceras:** para determinar el tratamiento de los fallos en esta etapa, se debe considerar el enfoque elegido para la comunicación intercluster. Si se adopta el enfoque sin agregación en la comunicación entre cabeceras, el nodo sink conoce la cantidad de mensajes que le deben llegar. Mientras la pérdida de mensajes no sea superior a un porcentaje que se fijará de acuerdo a la aplicación no toma ninguna acción, en caso que se supere ese porcentaje se considerará que el tiempo de vida de la red ha expirado. Si se adopta el criterio con agregación en la comunicación entre cabeceras, será cada CH el que deba verificar el porcentaje de pérdida de mensajes aceptado y tomar las acciones correspondientes. Esto es posible ya que cada cabecera conoce la cantidad de CH que reportan a él.

5 CONCLUSIONES

Se ha propuesto un algoritmo de medición periódica de las variables de interés. Este tipo de recolección de datos es muy utilizado en supervisión ambiental. En estas aplicaciones es muy frecuente que exista una fuerte correlación espacio/temporal de las variables involucradas, lo que permite utilizar técnicas de agregación. Se plantea el uso de una cuadrícula para evitar subdivisiones de la red y minimizar la cantidad de saltos para los mensajes de datos. Además, la propuesta de rotación de cabeceras de cluster contribuye al logro de un consumo más uniforme de potencia, prolongando el tiempo de vida de la red.

El algoritmo no garantiza la recepción de todos los mensajes en la estación base. Sin embargo controla dos puntos críticos en el envío, que son la selección de cabeceras y la definición de la estructura de comunicación entre cabeceras. Este control implica la incorporación de mensajes que incrementan el tráfico en la red. Dicho incremento se ha considerado necesario ya que los fallos en estas etapas degradan rápidamente el funcionamiento del sistema.

6 TRABAJO FUTURO

El funcionamiento del algoritmo propuesto debe ser verificado por simulación. Se espera realizar esta tarea con el fin de evaluar su desempeño para cotejarlo con el análisis realizado. Para un mismo entorno de trabajo se propone comparar la eficiencia del algoritmo propuesto en cuanto al tiempo de vida de la red, respecto a un algoritmo que no realiza ningún tipo de agregación. Otra simulación de interés es comparar los dos enfoques planteados, para la estructura de comunicación de cabeceras. Finalmente, se podría variar las dimensiones de la cuadrícula definida, lo que modificaría la definición de los clusters, y considerar su impacto sobre el desempeño de la red. Esta última prueba sería particularmente interesante cuando se trabaja con redes de distribución no homogénea de nodos.

REFERENCIAS

[1] Aboelaze M., Aloul F. Proceedings of Wireless and Optical Communications

Networks, 2005. WOCN 2005, 2005.

[2] Cook D., Das S. Smart Environments: Technologies, Protocols, and Applications. Wiley-Interscience, ISBN 0-471-54448-5, 2004.

[3] Krishnamachari B., Estrin D. and Wicker S. The Impact of Data Aggregation in Wireless Sensor Networks. *Proceedings of 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW '02)* p. 575, 2002.

[4] Dunlap R. In-Network Aggregation in Wireless Sensor Networks. College of computing. Georgia Institute of Technology. USA, 2004

[5] Karl H.,Willig A. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons. ISBN 0-470-09510-5, 2005.

[6] Shrivastava N., Buragohain C., Agrawal D. Medians and Beyond: New aggregation Techniques for Sensor Networks. University of California. *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, 2004. http://www.cs.virginia.edu/~son/cs851/papers/ucsb.sensys04.pdf

[7] Solis I. and Obraczka K. In-network aggregation trade-offs for data collection in wireless sensor networks. *International Journal of Sensor Networks 2006*, Vol. 1, No.3/4 p. 200 – 212, 2006.

[8] Al-Karaki J. N., Kamal A. E. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 2004

[9] AI-Karaki J. N., UI-Mustafa R., Kamal A E.. Data Aggregation in Wireless Sensor Networks - Exact and Approximate Algorithms. *Workshop on High Performance Switching and Routing*, p. (4) 241-245, 2004.

[10] Heinzelman W.. Application-specific protocol architectures for wireless networks. *In PhD thesis, Massachusetts institute of technology*, June 2000.

[11] D´Agostino E., Giandoménico E., Corti R., Martinez R. Ruteo en redes inalámbricas de sensores inteligentes, Universidad Nacional de Rosario, *Reporte técnico*, RT-ID-06/03, 2007. http://www.fceia.unr.edu.ar/secyt/rt/2006/rtid06_03.pdf

An improved convergence algorithm to compute $ln(x)$ – FPGA implementations

Géry J. A. Bioul, Martín Vázquez, Héctor N. Acosta, Martín Oriol*
Universidad Fasta, Facultad de Ingeniería
Gascón 3145 – B7600FNK – Mar del Plata. Argentina.
www.ufasta.edu.ar

Abstract

This paper presents FPGA implementations of classical algorithms for computing $ln(x)$ with some improvement at the level of the multiplication steps, and step skipping techniques. One starts from a practical implementation of $ln(x)$ computation using a convergence method. The function is approximated by a multiplicative normalization technique, however, thanks to the peculiarity of the multiplicative factor, namely $(1 + a_i .2^{-i})$, with $a_i \in \{-1, 0, 1\}$, the successive multiplications have been replaced by additions. Doing so, one saves the use of LUT's and eventually reduces processing time, as addition is generally faster than multiplication. Further, the acceleration technique, based on skipping trivial steps, improves performances. Implementations for FPGA are presented with time and slice cost evaluations. The Xilinx Virtex IV has been used for comparative analysis of 8 to 64-bit logarithm computing devices.

Keywords: FPGA, $ln(x)$, convergence method, multiplicative normalization, Xilinx Virtex IV.*

* This project is supported by FASTA University, Faculty of Engineering, B7600FNK Mar del Plata, Argentina.

1. INTRODUCTION

Most often the computation of functions such as logarithms, and exponential or trigonometric functions are made through software-implemented algorithms applied to floating-point representations. Hardware or micro-programmed systems are mainly justified for special-purpose computing devices such as ASIC or embedded systems. As it is generally not possible to get an exact result, approximation methods have to be used together with error estimation techniques. Newton-Raphson, Goldschmidt algorithm, Taylor MacLaurin series or Polynomial approximations are the most common approaches to compute these functions. For trigonometric functions, *CORDIC* (linear convergence) algorithms are well suited. Arguments included in the range [1, 2[- floating-point IEEE standard - are suitable for most approximation methods that need to limit the range of the argument. Whenever a specific range is imposed on the operands, a pre-scaling operation may be necessary: so an initial step may be included in the algorithmic procedure. Crucial questions for approximation methods are error estimation and effective rounding techniques; these problems start from tables design (first approximation LUT) up to the final result. Numerous methods, algorithms and implementations are proposed in the literature [1, 2, 3, 4, and 5]; the choice will depend upon the speed/cost compromises and other constraints imposed on the designer. Approximations methods usually assume available the four basic operations as arithmetic primitives at hand, together with look-up tables for a first "reasonably good" approximation to start from. This paper presents a practical implementation of $ln\ (x)$ computation using a convergence method [2]. The function is approximated by a multiplicative normalization technique, however, thanks to the peculiarity of the multiplicative factor, namely $(1 + a_i .2^{-i})$, with $a_i \in \{-1, 0, 1\}$, the successive multiplications have been replaced by additions. Doing so, one saves the use of LUT's and eventually reduces processing time, as addition is generally faster than multiplication. Furthermore an acceleration technique, based on skipping trivial steps, has been taken into account to improve performances. Implementations for FPGA are presented with time and slice cost evaluations. The Xilinx Virtex IV [6, 7, 8, 9] has been used for comparative analysis of 8 to 64-bit logarithm computing devices.

2. THEORETICAL BACKGROUND - LOGARITHM FUNCTION APPROXIMATION BY A CONVERGENCE METHOD USING MULTIPLICATIVE NORMALIZATION

Convergence methods consist in two parallel processes on two related sequences; typically, one sequence converges to 1 (*multiplicative normalization*) or 0 (*additive normalization*) while the other one converges to the function to approximate. Division using *Goldschmidt's* algorithm is an example of multiplicative normalization: while the divisor sequence converges to 1, the dividend converges to the desired quotient.

Define

$$c(i) = 1 + a_i .2^{-i}, \quad a_i \in \{-1, 0, 1\} \tag{1}$$

as the *multiplicative normalizing function*, where a_i is selected in such a way that the sequence

$$x(i+1) = x(i).c(i) \text{ (}auxiliary\ sequence\text{)} \qquad x(i) \in B(2^n) \tag{2}$$

converges towards 1. Then, the sequence

$$y(i+1) = y(i) - ln\ c(i) \tag{3}$$

can be set to converge toward the result $ln\ (x)$. If $y(0)$ and $x(0)$ are respectively set to 0 and to the argument x, and assuming $x(p) \cong 1$, one can write

$$x(p) = x.\Pi_i\, c(i) \cong 1 \rightarrow 1/x \cong \Pi_i\, c(i);\ \ y(p) = y - \Sigma_i\, ln\ c(i) = -\ ln\ \Pi_i\, c(i) = ln\ (x). \quad (4)$$

To make the convergence of (2) possible, the argument x needs to be in a range such that

$$x.min\ (lim_{p \rightarrow \infty} \Pi_{1 \le i \le p}\, c(i)) \le 1 \quad \text{and} \quad x.max\ (lim_{p \rightarrow \infty} \Pi_{1 \le i \le p}\, c(i)) \ge 1$$

that is

$$x \le 1/lim_{p \rightarrow \infty} \Pi_{1 \le i \le p}\ (1 - 2^{-i}) \text{ and } x \ge 1/lim_{p \rightarrow \infty} \Pi_{1 \le i \le p}\ (1 + 2^{-i}),\ \text{that is}\ \ 0.42 \le x \le 3.45. \quad (5)$$

This means that the argument x could need to be pre-scaled to fall in the range (5). An argument x in the range [1, 2[(such as e.g. a floating-point mantissa) fits perfectly; otherwise use a straightforward pre-scaling operation that replaces x by x' such that $x = x'.2^s$ (x' in [1, 2[); the algorithm computes $ln\ (x')$, then a final additive correction of $s.ln\ (2)$ is completed. Observe that the lower bound of (5) can be lowered to 0.21, as $(1+2^0)$ can be accepted as a first normalizing factor for computing $x(1)$.

In practical implementations of this algorithm, look-up tables are used to read out the successive values of $ln\ (1 \pm 2^{-i})$, needed to compute $y(i+1)$ of (3). For x in [½, 2[, a_i can be selected according to the following rules:

$$a_0 = 0, \quad (6)$$

$$\text{if } x(i) > 1,\ \ a_i = -\ x_{-i}\ (i), \qquad i \ge 1 \quad (7)$$

$$\text{if } x(i) < 1,\ \ a_i = +\ x_{-i}(i).not\ (x_{-i-1}(i)). \qquad i \ge 1 \quad (8)$$

The above rules are justified by the following two lemmas, also showing that the convergence rate reaches precision p after p steps (linear convergence).

Lemma 1.

Let

$$x(k) = 1 + 2^{-k} + \varepsilon, \qquad 0 \le \varepsilon \le 2^{-k} - 2^{-n}, \quad k \le n, \quad (9)$$

be the n-bit auxiliary sequence vector at step k; then

$$1 - 2^{-2k} \le x(k).(1 - 2^{-k}) < 1 + 2^{-k}. \quad (10)$$

Proof

The left inequality is trivial, it corresponds to $\varepsilon = 0$. The right inequality is deduced from the computation of $x(k).(1 - 2^{-k})$ for ε maximum, i.e. $2^{-k} - 2^{-n}$.

The practical interpretation of (10) is the impact of rule (7) on $x(k+1)$ whenever $x(k)$ is greater than one with a fractional part made up of a $(k-1)$-zero string and a one at position k. $x(k+1)$ will be either greater than one, exhibiting a similar pattern with at least one zero more, or inferior to one ($x_0(k+1) = 0$) with at least $2k$ one's as the header of the fractional part. In both cases, the target value $x(p) = 1$ is approximated by $x(k+1)$ with at least one bit more.

Lemma 2.

Let

$$x(k) = 1 - 2^{-k} + \varepsilon, \qquad 0 \le \varepsilon \le 2^{-k} - 2^{-n}, \quad k \le n, \tag{11}$$

be the n-bit auxiliary sequence vector at step k, then

$$1 - 2^{-2k} \le x(k).(1 + 2^{-k}) < 1 + 2^{-k}. \tag{12}$$

Proof

The right inequality is trivial, it corresponds to $\varepsilon = 0$. The left inequality is deduced from the computation of $x(k).(1 + 2^{-k})$ for ε maximum, i.e. $2^{-k} - 2^{-n}$.

The practical interpretation of (12) is the impact of rule (8) on $x(k+1)$ whenever $x(k)$ is less than one with a fractional part made up of a k-one string and a zero at position $k+1$. $x(k+1)$ will be either less than one, exhibiting a similar pattern with at least $2k$ one's as the header of the fractional part, or greater than one ($x_0(k+1) = 1$) with at least $k+1$ zero's as the header of the fractional part. In both cases, the target value $x(p) = 1$ is approximated by $x(k+1)$ with at least one bit more.

3. COMMENT

1 The selection (6) is justified by the fact that a decision about multiplying by $a_i .2^{-i} + 1$ (1) cannot be made before knowing the next bit. Actually, considering bit x_0 only (either 1 or 0) one cannot know whether the sequence $x(i)$ is already 1 (end of convergence process) or not.

2 When $x(i) > 1$, the strategy described by (7) consists in detecting the first non-zero bit of $x(i)$ then multiplying by $(-2^{-i} + 1)$. When $x(i) > 1$, lemma 1 shows that, at step i, bits $x_{-k > -i}$ (i) are all zero's.

3 When $x(i) < 1$, the strategy described by (8) consists in detecting the last non-zero bit of $x(i)$ then multiplying by $(2^{-i} + 1)$. When $x(i) \le 1$, lemma 2 shows that, at step i, bits $x_{-k > -i}$ (i) are all one's.

4. ALGORITHMS

4.1. Algorithm 1 - Logarithm computation by multiplicative normalization

The argument x is in [½, 2[: $x = x(0).x(1)\ x(2) \ldots x(n)$. Let $xx(i,j)$ be the component j of $xx(i) = xx(i,0).xx(i,1)\ xx(i,2) \ldots xx(i,n)$. Let $lut(i) = ln\ (1+a(i).2^{-i})$ read from the table.

a(0):= 0; c(0):= 1; xx(1):= x; yy(1):= 0;
for** i **in** 1 .. p-1 **loop

```
if xx(i)= 1 then exit; end if;
if xx(i)> 1 then a(i):= -xx(i,i) else a(i):= xx(i,i)*not(xx(i,i+1)); end if;
c(i):= 1+a(i)*2**(-i); xx(i+1):= xx(i)*c(i); yy(i+1):= yy(i)-lut(i);
end loop;
```

4.2. Algorithm 2 - Logarithm computation by multiplicative, one-shift and add, normalization

The argument x is in [½, 2[: $x = x(0).x(1)\ x(2) \ldots x(n)$. Let $xx(i,j)$ be the component j of $xx(i) = xx(i,0).xx(i,1)\ xx(i,2) \ldots xx(i,n)$. Let $lut(i) = ln\ (1+a(i).2^{-i})$ read from the table.

```
a(0):= 0; xx(1):= x; yy(1):= 0;
for i in 1 .. p-1 loop
    if xx(i)= 1 then exit; end if;
    if xx(i)> 1 then a(i):= -xx(i,i) else a(i):= xx(i,i)*not(xx(i,i+1)); end if;
    xx(i+1):= xx(i)+a(i)*xx(i)*2**(-i); yy(i+1):= yy(i)-lut(i);
    end loop;
```

4.3. Example 1

In the following example the auxiliary sequence x(i) is computed in the binary system, while, for readability, the sequence $y(i)$ is computed in decimal; the precision is then readily verified. The functional values $ln\ (1\pm2^{-i})$ are assumed given by look-up tables. x is in [1, 2[.

Let

$x = x(0) = x_0.\ x_{-1}\ x_{-2}\ x_{-3}\ x_{-4}\ x_{-5} = 1.10111 = (1{,}71875)_{10}$

$y(0) = 0$

Compute $ln\ (x)$ with precision $p = 8$

i	a_i	$c(i)$ $a_i.2^{-i} + 1$	$x\ (i+1)$ $x(i).c(i)$	$ln\ c(i)$	$y(i+1)$ $y(i) - ln\ c(i)$
-	-	-	$x(0) = 1.10111$	-	$y(0) = 0$
0	$a_0 = 0$	$0.2^{-0} + 1$ $c(0) = 1$	(1.10111).(1) $x(1) = 1.1011100$	0	0
1	$a_1 = -1$	$-2^{-1} + 1$ $c(1) = 0.1$	(1.1011100).(0.1) $x(2) = 0.11011100$	-0.69314718	0.69314718
2	$a_2 = 1$	$2^{-2} + 1$ $c(2) = 1.01$	(0.11011100).(1.01) $x(3) = 1.00010011$	0.223143551	0.470003628
3	$a_3 = 0$	$0.2^{-3} + 1$ $c(3) = 1$	(1.00010011).1 $x(4) = 1.00010011$	0	0.470003628
4	$a_4 = -1$	$-2^{-4} + 1$ $c(4) = 0.1111$	(1.00010011).(0.1111) $x(5) = 1.00000010$	-0.064538521	0.534542149
5	$a_5 = 0$	$0.2^{-5} + 1$ $c(5) = 1$	(1.00000010).1 $x(6) = 1.00000010$	0	0.534542149
6	$a_6 = 0$	$0.2^{-6} + 1$ $c(6) = 1$	(1.00000010).1 $x(7) = 1.00000010$	0	0.534542149
7	$a_7 = -1$	$-2^{-7} + 1$ $c(7) = 0.1111111$	(1.00000010).(0.1111111) $x(8) = 1$ (rounded up)	-0.007843177	0.542385326

The actual decimal value of *ln* (1.71875) is 0.541597282 ± 10^{-9}, the difference from the computed result is less than $8.10^{-4} < 2^{-10}$.
As it appears in the preceding example, whenever $a_i = 0$, the only effect of step *i* on the computation process consists in incrementing the step number; both sequences $x(i)$ and $y(i)$ remain unchanged. So, by detecting strings of 0 or 1 in $x(i)$, one could readily jump to the next non trivial computation step. The following example illustrates this feature.

4.4. Example 2

As in the preceding example 1, the auxiliary sequence x(*i*) is computed in the binary system, while sequence $y(i)$ is computed in decimal. The functional values $ln\ (1 \pm 2^{-i})$ are given by look-up tables. x is now in [½, 2[. Strings **00..** or **11..** are highlighted. The multiplications by (1 ± 2^{-i}) have been replaced by additions: $x(i) \pm x(i)*2^{-i}$.

Let

$x = x(0) = x_0.\ x_{-1}\ x_{-2}\ x_{-3}\ x_{-4}\ x_{-5} = 0.10011 = (0{,}59375)_{10}$

$y(0) = 0$

Compute *ln* (x) with precision $p = 10$

i	a_i	$c(i)$ $a_i.2^{-i} + 1$	$x\ (i+1)$ $x(i).c(i)$	***ln c*(*i*)**	$y(i+1)$ $y(i) - ln\ c(i)$
-	-	-	$x(0) = 0.1001100000$	-	$y(0) = 0$
0	$a_0 = 0$	$0.2^{-0} + 1$ $c(0) = 1$	$(0.10011).(1)$ $x(1) = 0.1001100000$	0	0
1	$a_1 = 1$	$2^{-1} + 1$ $c(1) = 1.1$	$(0.1001100000).(1.1)$ $x(2) = 0.1001100000$ $+ 0.0100110000$ $= 0.1\mathbf{11}0010000$	0.405465108	0.405465108
2	$a_2 = 0$	-	- $x(3) = x(2)$	-	-
3	$a_3 = 1$	$1.2^{-3} + 1$ $c(3) = 1.001$	$(0.111001).(1.001)$ $x(4) = 0.111001$ $+ 0.000111001$ $= 1.000\mathbf{00000}10$	0.117783035	0.523248143
4→8	$a_4 \rightarrow a_8$ $= 0$	-	- $x(9) = x(4)$	-	-
9	$a_9 = -1$	$- 2^{-9} + 1$ $c(9) =$ 0.111111111	(1.0000000010) $.(0.111111111)$ $x(10) = 1.0000000010$ $- 0.0000000010\ ..$ $= 1$ (rounded up)	- 0.001955035	0.521293108

The actual decimal value of *ln* (0.59375) is - 0.521296923 ± 10^{-9}, the difference from the computed result is less than $4.10^{-6} < 2^{-10}$.

4.5. Implementation schemes

4.5.1. Algorithm 1

Figure one displays a possible implementation scheme for algorithm 1. The auxiliary sequence is computed through successive multiplication by 1, $(1+2^{-i})$ or $(1-2^{-i})$ according to the values of a_i .

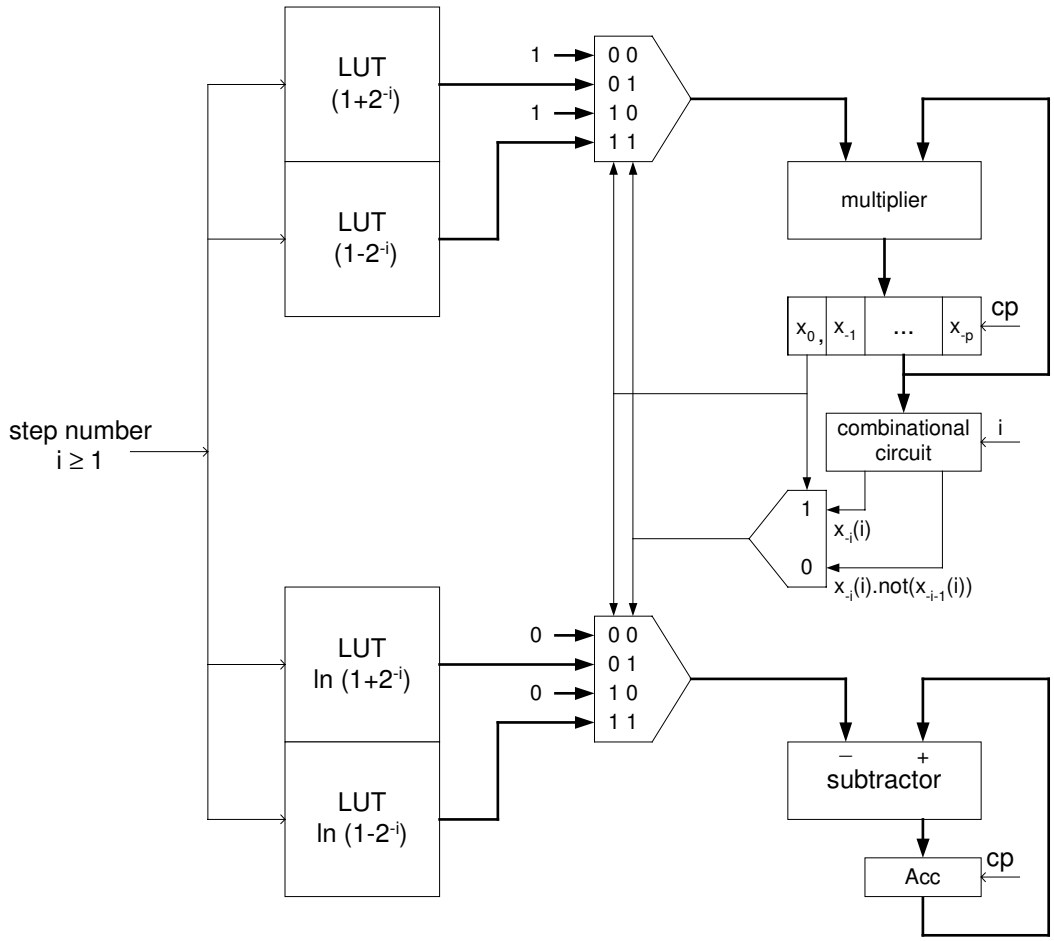


Figure 1. Logarithm computation circuit using multiplicative normalization

Figure 2 displays a possible implementation scheme for algorithm 2. The auxiliary sequence is computed through successive additions of 0, $x(i)*2^{-i}$ or $-x(i)*2^{-i}$, according to the values of a_i . Actually the algorithm materialized by figure two is a slight modification of algorithm 2, as follows.

4.5.2. Modified algorithm 2

a(0):= 0; xx(1):= x; yy(1):= 0;

```
for i in 1 .. p-1 loop
    if xx(i)= 1 then exit; end if;
    if xx(i)> 1 then a'(i):= xx(i,i); xx(i+1):= xx(i)-a'(i)*xx(i)*2**(-i);
    else a'(i):= xx(i,i)*not(xx(i,i+1)); xx(i+1):= xx(i)+a'(i)*xx(i)*2**(-i); end if;
    yy(i+1):= yy(i)-lut(i);
    end loop;
```

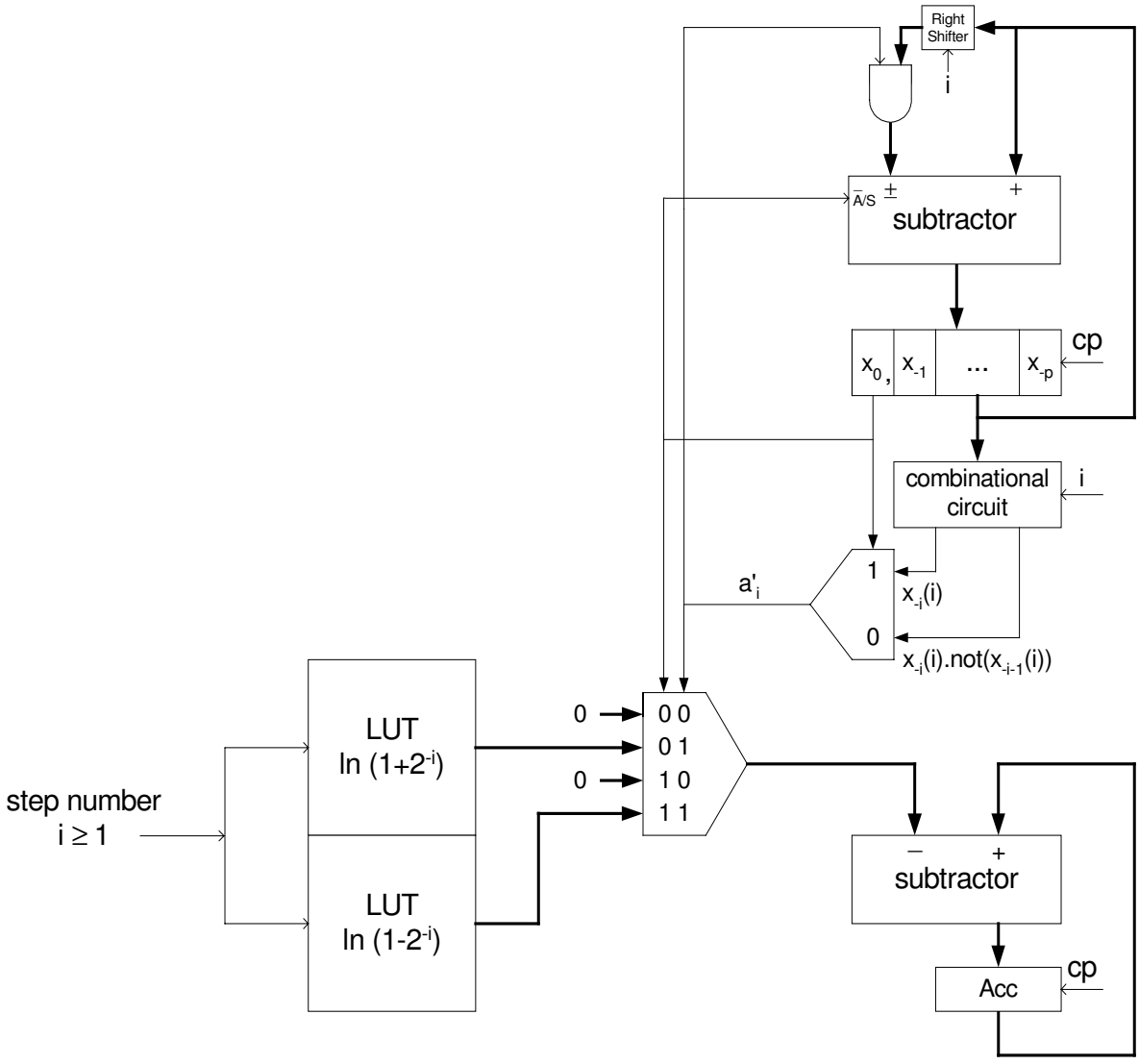


Figure 2. Logarithm computation circuit using a multiplicative normalization circuit made of a shifter and an adder/subtractor

5. IMPLEMENTATIONS ON FPGA XILINX VIRTEX IV

5.1. Algorithms 1 and 2

Both *ln*(x) algorithms presented above have been implemented on a 10752-slice Virtex IV FPGA [6]. Synthesis has been achieved using Xilinx Synthesis Technology-*XST*, [7], while physical implementation used Integrated System Environment (ISE), [8]. A comparative analysis is presented for 8-bit, 16-bit, 32-bit and 64-bit *ln*(x) precisions. The corresponding precision is provided by the respective LUT's.
For implementing algorithm 1, among the available synthesis options, DSP48 units [9] have been selected for efficiency purposes. DSP48 is a dedicated unit involving an 18-bit multiplier, a 3-input 48-bit adder together with the corresponding additional logic such as multiplexers. The aimed functional precision determines the required quantity *ND* of DSP48 unit, namely

$$ND = (N/16)^2, \qquad (13)$$

where *N* stands for *N*-bit precision.

Table 1 displays the comparative values of cycle time *T* (single-step time), operating frequency *F*, *ln*(*x*)-operation frequency *FOP*, number of required DSP's *ND*, and number of slices *NS* for 8-bit, 16-bit, 32-bit and 64-bit precision and for implementations of algorithms 1 and 2.

Table 1: Performances of algorithms 1 and 2 implemented on Xilinx Virtex IV (device xc4vlx25-12ff668)

N	**Algorithm 1 (multiplier)**						**Algorithm 2 (one-shift & Add)**				
	T *(ns)*	***F*** *(Mhz)*	***FOP*** *(Mhz)*	***ND***	***NS***		***T*** *(ns)*	***F*** *(Mhz)*	***FOP*** *(Mhz)*	***ND***	***NS***
8 bits	6.2	161	23.0	1	44 (0.4%)		4.5	222	31.746	-	52 (0.5%)
16 bits	6.7	149	9.95	1	92 (0.9%)		5.8	172	11.494	-	115 (1%)
32 bits	12.9	77	2.50	4	221 (2%)		7.2	138	4.480	-	290 (2.7%)
64 bits	20	50	0.793	16	639 (5.9%)		8.3	120	1.912	-	707(6.6%)

ln(x) operation frequency FOP is computed as

$$FOP = F/(N-1) \qquad (14)$$

Table 2 enhances the improvements of *FOP*'s for Algorithm 2 with respect to Algorithm 1 and the related increases of slice costs. It can be observed that the operating speed-up is paid by some additional slice cost but for $N = 32$ and $N = 64$, the overall performance is reached at a very reasonable cost. It can be observed that the improvements are more significant for higher values of *N*, due to the quadratic increase of needed DSP48's.

Table 2: Comparative *FOP* and slice costs of algorithms 1 and 2 implementations

N	***FOP*(2) / *FOP*(1)**	***NS*(2) / *NS*(1)**
8 bits	1.38	1.18
16 bits	1.15	1.25
32 bits	1.79	1.31
64 bits	2.41	1.11

5.2. Algorithm 3 - acceleration of algorithm 2.

An important feature of the *ln*(x) algorithms presented in section 4, is the possibility to skip steps according to eventual values zero of a_i's. As a matter of fact whenever $a_i = 0$, neither *c(i)* nor *y(i)* have to be modified. According to rules (7) and (8), one can skip as many steps (say *s*) between position *-i* and position *-i-s*, as allowed by the length of eventual 0-strings (within *x(i)*>1) or 1-strings (within *x(i)*<1). One can shows statistically that the average quantity of steps to be skipped, using this acceleration procedure, is superior to 50 % of what would be required by the straight *p*-step procedure for precision *p* calculation. Exhaustive tests have been handled up to 16-bit operands, while statistic experiments (for normalized [1,2[operands) exhibited Gaussian distributions. Table 3 displays the experimental average latency (***L***) in terms of the average numbers of required cycles for 8-bit, 16-bit, 32-bit and 64-bit operands. The drawback of step skipping is the rise of cycle length, due to the need of an additional circuit to set the skip length (*s*). Actually the hardware at hand plays a key role in additional costs and subsequent performance improvements. Table 4 displays the overall performances of the accelerated algorithm 2 while table 5 displays the comparative *FOP* and slice costs of algorithm 2 with respect to the accelerated version (algorithm 3).

Table3: Average latency for algorithm 3

N	***L***
8 bits	3.4
16 bits	7.1
32 bits	15.1
64 bits	31.1

Table 4: Performances of algorithm 3 implemented on Xilinx Virtex IV (device xc4vlx25-12ff668)

	Accelerated algorithm 2 (Algorithm 3)			
	T *(ns)*	***F*** *(Mhz)*	***FOP*** *(Mhz)*	***NS***
8 bits	6.4	156	45.955	59 (0.5%)
16 bits	8.9	112	15.825	147 (1.4%)
32 bits	11.5	86	5.758	397 (3.7%)
64 bits	16.3	61	1.978	1217 (11.3%)

Table 5: Comparative *FOP* and slice costs of algorithms 3 and 2 implementations.

N	***FOP*(3) / *FOP*(2)**	***NS*(3) / *NS*(2)**
8 bits	1.45	1.13
16 bits	1.37	1.27
32 bits	1.28	1.38
64 bits	1.03	1.72

Table 5 shows that the performances (*FOP*) are decreasing as *N* increases. This means that as *N* increases the additional step delay is overcoming the reduction in the number of steps. The *NS*

factor is directly related to the step complexity. In order to take a better profit of the acceleration technique, some optimization technique is needed and remains an open question.

6. CONCLUSION

Starting from a classical $ln(x)$ computation algorithm using convergence method with multiplicative normalization, some FPGA implementations have been carried out. The used FPGA device belongs to the Xilinx Virtex4 family.
On first noticed that the multiplication steps may be usefully replaced by a shift and add procedure using a shifter and an adder-subtractor. This alternative has proved to be faster and cheaper. To emphasize this point, one first implemented algorithm 1 (multiplication) using dedicated DSP48 multiplier cells embedded in the device at hand. Then the algorithm 2 (shift and add) has been implemented and provided up to 140 % performance improvements (*FOP*) - for 64-bit operand, while the hardware cost augmented by 11 % only. Finally the acceleration of the process (skipping trivial steps) has been taken into account to improve performances, but the reduction in number of steps appeared to be partially compensated by an additional delay generated by step length. So, the advantages vanish as N increases.

7. BIBLIOGRAPHY

[1] J. Cao, B.W. Wei, and J. Cheng, "High-Performance Architecture for Elementary Functions Generation," *Proc. 15th IEEE Symp. Computer Arithmetic,* pp.136-144, 2001.
[2] J-P. Deschamps, G. Bioul, and G. Sutter, *Synthesis of Arithmetic Circuits, FPGA, ASIC, and Embedded Systems,* John Wiley Interscience, New York 2006.
[3] M.D. Ercegovac, "FPGA Implementation of Polynomial Evaluation Algorithms," *Proc. of SPIE Photonics East '95 Conference,* Vol. 2607, pp177-188,1995.
[4] V. Paliouras, K. Karagianni, and T. Stouraitis, "A Floating-point Processor for Fast and Accurate Sine/Cosine Evaluation," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal processing,* Vol. 47 n°5, pp. 441-451, May 2000.
[5] P.K. Tang, "Table Look-up Algorithms for Elementary Functions and their Error Analysis," *Proc. 10th IEEE Symp. Computer Arithmetic,* pp.232-236, 1991.
[6] Xilinx inc., Virtex-4 User Guide, http://www.xilinx.com, April 2007.
[7] Xilinx inc., XST User Guide-82i, http://www.xilinx.com, 2007.
[8] Xilinx inc., ISE 8.2 documentation, http://www.xilinx.com, 2007.
[9] Xilinx inc., Xtreme DSP for Virtex-4 FPGA's User Guide, http://www.xilinx.com, June 2007.

2007 Argentine Congress on Computer Science (Congreso Argentino en Ciencias de la Computación - CACIC 2007)

An Overview of MPLS Technology: Quality of Service and Traffic Engineering

Rubén J. Fusario, E. Carrara and J. Mon
Departamento Sistemas, Facultad Regional Buenos Aires,
Universidad Tecnológica Nacional, Buenos Aires, Argentina
rfusario@speedy.com.ar, emacarrara@gmail.com, jorgemon@speedy.com.ar,

and

Antonio Ricardo Castro Lechtaler
Universidad de Buenos Aires and Departamento Sistemas, Facultad Regional Buenos Aires,
Universidad Tecnológica Nacional, Buenos Aires, Argentina
acastro@utn.edu.ar

and

Carlos García Garino
Carrera Redes y Telecomunicaciones, ITU and Facultad de Ingeniería
Universidad Nacional de Cuyo; Mendoza, Provincia de Mendoza
cgarcia@itu.uncu.edu.ar

Abstract

This paper provides an overview of Multi-Protocol Label Switching Technology (MPLS). Distinctive issues are presented as well as Quality of Service (QoS). Traffic Engineering (TE) features are discussed. It also describes current network trends where Internet Protocol (IP) is the predominant technology and points out the advantages of MPLS for data, voice, and video information transport.

Keywords: Multi Protocol Label Switching, QoS, Traffic Engineering

1. INTRODUCTION.

IP is currently the most widely used network protocol. It is designed on the so called best-effort idea and is non-connection oriented. The protocol has been successfully used. However, some shortcomings arise when IP Networks have to transport voice and video data. In these cases, lack of bandwidth, low latency, and loss of packages become important issues that need to be addressed.

Different technologies have been suggested to circumvent the cited problems. Quality of Services (QoS) based in standards like Differentiated Services (DiffServ) [1] and Integrated Services (IntServ) [2] are among the most suitable ones. Asynchronous Transfer Mode (ATM) [3] has been widely used as a backbone technology by carriers but it's difficult to extend it to enhance network capillarity. In the last few years, Multi Protocol Label Switch (MPLS) [4-7] emerges as a promising technology. In this sense, QoS and Traffic Engineering (TE) [8-10] are capable of superseding IP Networks when MPLS is used.

QoS guarantees that a given data flow related to a specific application, such as VoIP, Video Conference, critical applications, and so on, has a differential treatment over standard traffic. In this way, each data flow is managed according to specific application requirements.

Traffic Engineering is a process that optimizes network resources. In order to achieve this objective, traffic flow is distributed in an independent way of standard layer 3 routing protocols. Resource optimization here means that different paths are assigned for different incoming protocols according to their distinctive features, instead of assigning the shortest path or the most economical route, for instance.

In section 2, QoS technologies used for network deployment are discussed. MPLS is introduced in section 3. Distinctive features are described in proper subsections before presenting examples in section 4. Conclusions are provided in section 5.

2. QoS PROCUREMENT

QoS issues on IP networks are addressed in subsection 2.1 and QoS strategies are discussed in subsection 2.2

2.1 Quality of Service on IP Networks

Providing service quality is an important necessity whenever network overload is present. To do so, certain mechanisms should be present to supply preferential treatment to specific packages from total traffic. This process would not be necessary if the network were to be oversized.

Agreements with providers must be established to ensure the range of certain parameters to bring about service quality. Service Level Agreement (SLA) establishes the QoS parameter values agreed upon between client and provider. *Availability*; *Data Package Loss*; *Bandwidth*; *Delay* and *Jitter* can be cited among the most important parameters

Availability denotes the percentage of time (usually in annual or monthly terms) that the provider ensures service availability, i.e., the server will be operative. Usual values: 99.7% - 99.9%
Data Package Loss account for the maximum quantity of lost data packages, considering hired bandwidth. Usual values are 0.1%.

The provider guarantees the client a certain *bandwidth* within its network. Usual Values: 5 Mbps. Average two-way package delay between origin and destination are generically denoted as *Delay* and its variation as *Jitter*. Usual Values for Delay and Jitter are 50 mseg and +/-10mseg, respectively.

Delay is originated by the nature of the environment (the different media in which data goes through), as well as by the quantity of hops between origin and destination. Jitter, on the other hand, results from the gridlock of data packages remaining in queue without being dispatched or having to take alternate routes to arrive to destination.

2.2 QoS Strategies.

Two possible strategies give specific data packages preferential treatment from the rest of the traffic: *Reservation* and *Priority*

Reservation denotes the case when communications capacity is reserved for exclusive use. Reservation should be carried out in every router up to destination, requiring a reservation protocol. Routers must keep traffic information. However, package identification is not necessary. Assurances are provided. The standard used in this strategy is Integrated Services (IntServ) [2].

Priority is the case when packages are marked according to specific characteristics such as origin, destination, port, and so on. These features allow routers to grant preferential treatment. It is a statistical treatment and thus cannot provide full guarantees.

3. MPLS DESCRIPTION.

The most widely used method to forward IP packages is based on IP head analysis, specifically, in the destination address field. It consists of a layer 3 search through an algorithm aiming to attain the next hop. This expensive process needs to be performed in every router between origin and destination.

In practice, different technologies have been suggested. In this work, the recent MPLS [4-9] case is discussed. An introductory overview of the technology is introduced in subsection 3.1. Then MPLS fundamentals, processing, and TE [10-12] issues are addressed in subsections 3.1, 3.2 and 3.3, respectively. The architecture of MPLS networks is described in subsection 3.4.

3.1 Introduction to MPLS Capabilities and Infrastructure.

It was pointed out in the introduction that multimedia traffic requires QoS not available in IP network infrastructure. In practice, different approaches have been followed to provide some extent QoS capabilities both for LAN and WAN infrastructure. These cases are used to introduce MPLS technology before discussing fundamentals, as well as QoS and TE issues in the following subsections.

Ethernet networks dominate LAN technologies. Several enhancements have been successfully applied in practice after the original, shared network, ruled by the IEEE 802.3 standard. It could be argued that Fast Ethernet and Switched Ethernet are the more remarkable ones [13]. However, QoS was still an option until VLANs [14] were introduced in practice.

Tagged frames were introduced in IEEE 802.1q which include some weak QoS capabilities. Other distinctive features are layer 2 forwarding packets. In the case of VLAN's, frame switching can be based on VLAN's tags, distinguishing core and out of core areas. Border switches add/eliminate tags from incoming/outcoming packages to communicate with legacy Ethernet infrastructure.

The migration from original to the switched VLAN's case requires updating of switching hubs but wiring practically remains unchanged, as well as upper layers are unaffected. Up to some extent,

VLANs can be considered like a local technology that incorporates several WAN features like full duplex links, switched circuits or layer 2 forwarding.

In the case of WANs, ATM networks consider QoS as a native feature. Moreover, the aware reader will recognize traffic categories like CBR, VBE-rt, ABR, UBR, and so on. Specifically designed, High Speed switched circuits and small size cells guarantee a high speed QoS featured technology. However, ATM is rather expensive and has been used in practice only for backbones.

Main carrier network infrastructure still relies on classical IP links that lack any QoS capabilities but could become problematic when managing multimedia traffic. In this context, MPLS becomes a bridge between high speed QoS featured links and legacy infrastructure.

From a conceptual point of view, standard frames are enhanced (see table 1) similar to 802.1q ones for the case of the Ethernet. The added information is considered to forward packages to destination. Consequently, new capabilities are added to MPLS aware router that forward packages based on a layer 2/layer 3 combination criteria. In his paper, Stallings details MPLS label inclusion in frames of different technologies. [5]

Based on MPLS information, different types of traffic can be forwarded according to their nature. In this sense, voice and video data have priority over batch frames like e-mail or regular internet traffic. While WAN network links remain unchanged, routing equipment has to be properly updated; for instance, in the VLAN's case.

MPLS technology has the advantage to be designed for WAN infrastructure and the capability to be deployed over traditional links like IP, Frame Relay, and ATM. SONET and so on, as shown in figure 1. Given the QoS capabilities of MPLS, practically any kind of services can be transported using this technology as indicated in figure 1.

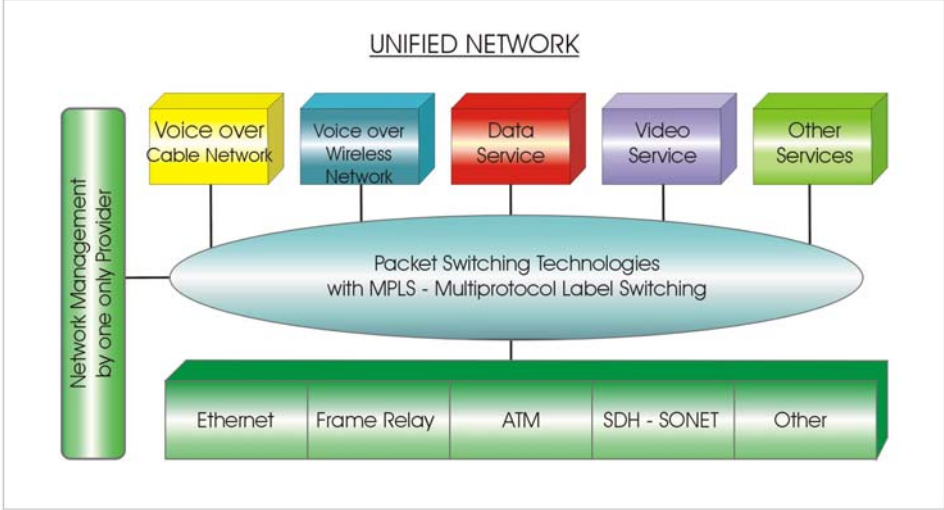


Figure 1: MPLS services and infrastructure

In summary, MPLS allows carriers to ensure traffic QoS levels proper of ATM links, but deployed, in this case, over legacy technologies like IP or Frame Relay. Consequently, carriers take advantage of their large IP/Frame Relay infrastructure, offering high capillarity to their networks. The added

value in this case is the QoS offered by MPLS equipment. The approach of MPLS from a conceptual point of view presents some loose analogies to the VLAN's case.

3.2 MPLS Fundamentals

MPLS is a new architecture fostering the use of labels to combine the layer 2 forwarding benefits with those of the layer 3 routing techniques. The labels are assigned to packages to be transported through package or cell based networks and the forwarding mechanism takes place with a label exchange. In table 1 can be seen an example of MPLS labels.

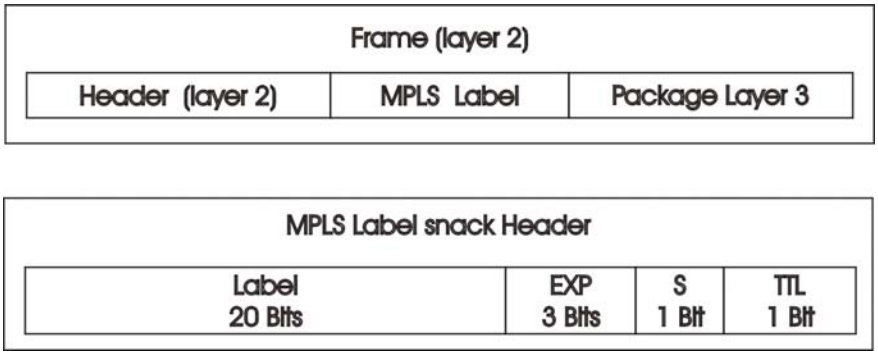


Table 1: MPLS Frame

This architecture is multi-protocol because it can be assigned to any layer 3 network protocol. However, the main focus is to use it on IP networks. The MPLS architecture can be modeled in two main components: Forwarding and Control [4]:

Forwarding Component is used to do the forwarding task based on labels which are inserted, replaced, and removed through this module. Insertion and removal are performed at the border nodes according to a database known as Forwarding Information Base (FIB). Replacement is performed on the central nodes according to a database known as Label Forwarding Information Base (LFIB).

Control Component deals with the gathering, maintenance and distribution of the necessary information to make decisions about forwarding. The information is stored in an additional database known as Label Information Base (LIB). From this database and the routing tables the FIB and LFIB are created.

MPLS nodes must run at least one routing protocol (BGP, OSPF, and others) to maintain the routing table and exchange information with other network nodes.

The routing table allows MPLS nodes to determine who their neighbors are to exchange label information, following known sub-networks. These sub-networks are registered in the routing table. The label exchange is accomplished through a protocol such as Label Distribution Protocol (LDP).

3.3 Data Forwarding

3.3.1 IP Data Forwarding

A data package whose network layer protocol is oriented to no-connection must travel from one node to another. Each node must make a forwarding decision independently. This decision is taken

through an analysis of the package head with a network layer algorithm. At this point, it is important to introduce the concept of *Forwarding Equivalence Classes* (FEC). A FEC describe a group of packages that are classified and forwarded in the same way. Consequently all the packages that belong to the same FEC follow the same path.

Two processes divide the next-hop (or node) choice in the path:

- The first process partitions the entire traffic of possible packages n an FEC set.
- The second process associates each FEC with the next hop or node; i.e., it selects a particular hop for each package group which compose FEC.

In the case of conventional IP networks, when analyzing heads, a group of packages is assigned to a specific FEC. Afterwards, each FEC is mapped to the next hop, but throughout the path, each router re-examines the packages and assigns another FEC. It is an effective process, although inefficient if required processing time for each package in every node is considered.

3.3.2 MPLS Data Forwarding

In the case of MPLS, a package is assigned to a FEC as soon as it enters the domain of a MPLS network through the allocation of a label. In the next hop, it is exchanged to a new one and sent to the next hop. The entire forwarding procedure, within the domain of the MPLS network, is carried out with these labels, avoiding the analysis of the entire package. It can be used in packages as well as cells. Standard routing techniques like Internet Gateway Protocols (IGP) are still necessary in order to exchange the information required to set up and update the MPLS database.

The path followed by packages in an MPLS cloud is denoted as Label Switch Path (LSP). Label Switch Routers (LSR) manage the creation, replacement and interchange of labels. A diagram of a MPLS network is shown in figure 3.

The allocation process of a FEC is carried out with an Edge-LSR which performs a network level search (checking with FIB) upon receiving a package to identify the IP address of the next hop.

If it detects that the next hop is accomplished through an interphase connected to the MPLS network, it performs a search in the FIB, obtaining a label associated to the destination network prefix. It inserts a label between the layer 2 head and the layer 3 package. It will then dispatch the package through the appropriate interphase.

The label exchange process is carried out in the intermediate LSR nodes, which upon reception of the labeled package, will search in their LFIB, the new label, and the corresponding interphase to that FEC. Afterwards, they will dispatch the resulting interphase with the new label replacing the previous one.

The label extraction process is carried out by an outgoing Edge-LSR. Upon reception of a labeled package from an MPLS device, the Edge-LSR searches in its LFIB the new label and corresponding interphase to that FEC. Upon detecting that the new label does not exist, as it belongs to an interphase connected to a no-MPLS device, it extracts the label and performs a search at the network layer level (looking up in the FIB) to identify the next hop IP address and dispatches to the appropriate interphase. LSPs are unidirectional. Thus, return traffic from a specific FEC is carried out on a different LSP.

Labeling packages indicating the paths that they can follow and guiding them through the network taking advantage of redundant paths will favor the flow of data. The chosen path may differ from the one established at level 3. Labeling or re-labeling processes (swapping) may be carried out in function of other variables besides addresses, enabling QoS application.

The difference of MPLS technology with traditional WANs lies in the way labels are applied and in the ability to create label stacks, enabling new applications such as traffic engineering, VPNs, rapid recovery of gridlocked routes, and QoS.

3.4 Traffic Engineering

Conventional traffic forwarding like the one following IGP protocols tends to follow the shortest path first (SPF) or the more economical routes. In this way, most of packages follow the same path. Then, traffic gridlock occurs while other routes have available bandwidth. In order to change this behavior, routing metrics should be changed. However, side effects usually take place. New techniques like Constraint Based Routing (CBR) have been suggested in practice to alleviate this problem [10]. This capability is one of the distinctive features of Traffic Engineering (TE) [10-12].

Figure 2 shows an example of alternate routes between two points (for example between node A and node B) and the main route. When the shortest path, ab in this case, is gridlocked, it might be useful to route traffic through ac-cb in order to reduce the ab gridlock. This choice is impossible to implement in practice with standards routing protocols pointed out in a previous paragraph.

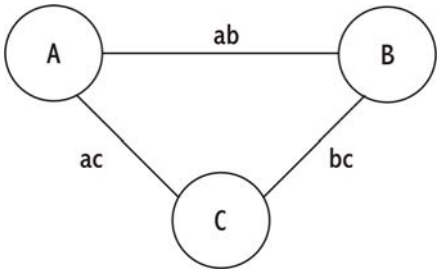


Figure 2: Alternative routing in TE

MPLS networks may use TE processes to minimize gridlock and increase network performance. Traffic Engineering changes the routing patterns to provide efficient mapping of traffic flows to the network resources. It may reduce gridlock and increase service quality in terms of latency, jitter, and package loss [23]. MPLS traffic engineering involves several considerations:

Information distribution over links uses IP routing protocol extensions of the link state to distribute information of the topology. LSR requires the information to transport based on restrictions. IS-IS or OSPF protocols distribute the necessary link attributes to form the topological database.

LSR performs *path calculations* using the information of the topological database (in particular, link attributes: managing groups, bandwidth, measures, etc.), through extensions of the SPF algorithm, called Constraint-based, shortest path first (CSPF).

In MPLS network Label Switch Path (LSP) is a common route for packages which belong to the same FEC. In order to provide *Path Signaling* capabilities, RSVP protocol extensions are used. Basically, a “Path-Label Request” message is sent, requiring the association of a label to a particular

path in each hop. Each node, beginning from destination, after receiving a message, distributes the labels to the closest node towards the origin using a RESV – Label message. At first, LDP extensions were used for the signaling (CR-LDP); but in 2002 IETF favored RSVPTE as MPLS TE signaling protocol.

Finally, it is necessary to provide *Traffic Selection* to define which packages are assigned to a particular LSP formed by related labels. TE LSP traffic must enter the first LSP node through a selection process which might be dynamic, static, type-based or content-based.

3.5 MPLS Architecture with QoS and TE

A typical MPLS architecture is depicted in figure 3. Customer and Provider Edge equipments, CE and PE respectively, as well as LSP are shown in the diagram.

CE (Customer Edge) commands package labeling performed in the field's origin IP address, destination IP address, priority, protocol, port of origin, port of destination, and so on. The information is gathered from the analysis of the IP head and the transported datagram (TCP, UDP, etc.). The label, shown in Table 1, is represented over a 3 bit priority with 8 possible codes (2^3). In DiffServ architecture, the field ToS is redefined in the field DS which contains the sub-field DSCP (Differentiated Services Code point), unifying the priority 3 bits D, T, and R, with 64 possible codes (2^6). The use of a 6-byte DSCP is purposeless considering that the EXP field from the MPLS head also has 3 bits.

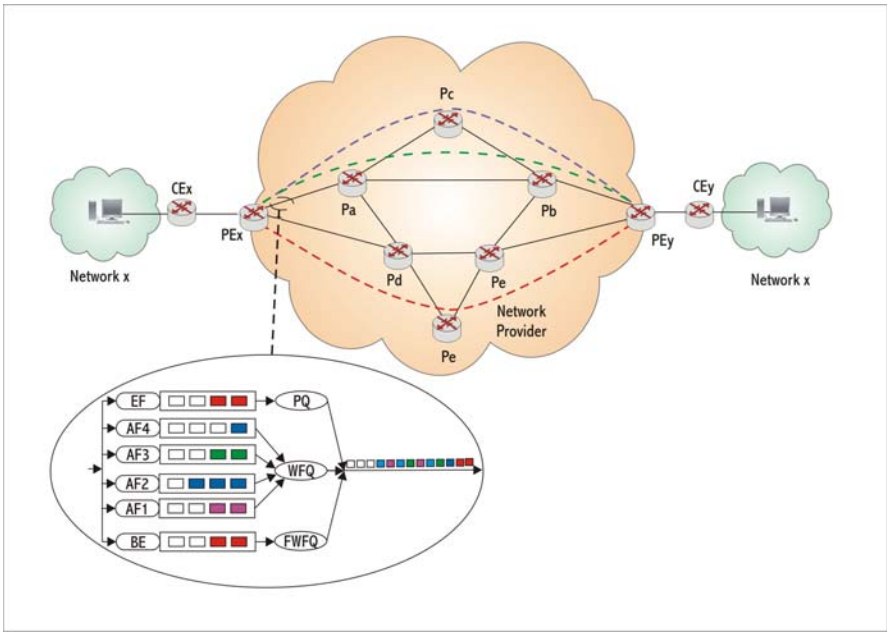


Figure 3: MPLS with QoS and TE Network Diagram

The data packages leave the CE with a new priority value in the field ToS and are directed to the next hop (PE – Provider Edge), whose address was obtained through the level 3 routing table lookup.

PE receives No-MPLS packages through the interphase connected to CE. Given that it is an Edge-LSR, it performs a layer 3 lookup, a FIB search, and identifies the label as well as the departure interphase – see section 3, MPLS Architecture, Processing. Afterwards, it classifies the packages and deposits them in the departure queues associated to the selected interphase (shaping).

In P nodes (Provider – Belonging to the Core), packages can be labeled or reclassified. In the case of existing alternate paths to a same destination, the associated interphase to each path can treat their queues differently.

4. EXAMPLES.

In this section, different MPLS applications are shown in order to present some practical implementations. The bandwidth requirement are computed in the example of subsection 4.1 and a VPN deployment is discussed in subsection 4.2

4.1 LAN Bandwidth requirements

The following steps are followed:

4.1.1. Design and install switched Ethernet LANs in the different sites of the corporative network. The following requirements are mandatory:

a. The ANSI/EIA/TIA 568 standards must be followed.
b. A domain server has to be installed at each site. An e-mail server can be optionally included.

4.1.2. Interconnect the different sites using an IP/MPLS transport network like a corporate backbone. A carrier provides the technology and links necessary to deploy the IP/MPLS facilities.

4.1.3. LAN's Bandwidth requirements are previously computed to allow for a proper Carrier design of the corporate backbone. From authors' experience, 10 kbps should be added for each PC (considering Internet and corporate applications); 25 kbps are advisable for each VOIP channel. If videoconference facilities are to be included, additional 256 kbps should be added. In this way, for a 20 PC site plus two VOIP channels as well as a videoconference link, the computed bandwidth is approximately 512 kbps.

4.2 VPN over MPLS deployment

The diagram in figure 4 shows a corporate WAN that is connected to Internet and has different remote VPN links.

In the graph, the IP/MPLS backbone provided by the Carrier can be seen. Different sites where one or more LAN’s are available (Fast Ethernet or GigaEthernet generally) are connected thorough a CE (Customer Edge) that is in charge to tag the packages as has been explained in the previous section.

Users properly authorized can access the backbone. VPN’s links are established in between user’s site and the access router to the IP/MPLS corporate backbone. In this way, Internet infrastructure can be used in order to set up secure connections.

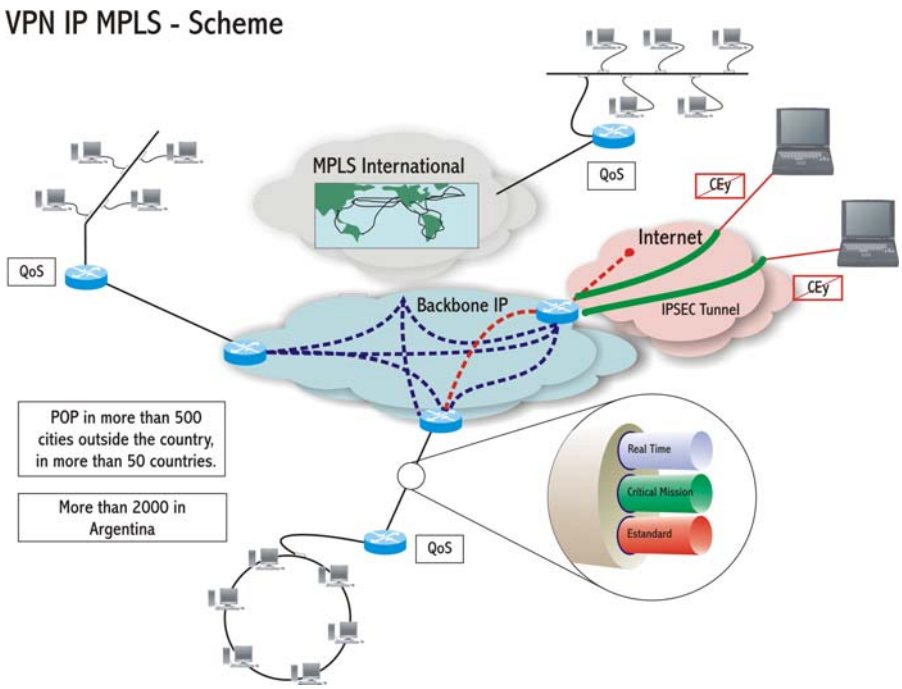


Figure 4: Example of IP/MPLS deployment.

Usually the corporation requires the Carrier the desired bandwidth. In turns, this bandwidth can be classified according QoS necessities. For instance, given 1 Mbps access 30 % are guaranteed for VoIP purposes, 40 % for critical mission traffic and the remaining bandwidth is available for internet and e-mail traffic.

5. MPLS IN ARGENTINA.

The most important carriers in Argentina currently offer IP/MPLS links to their customers [15]. Generally, when international connection is available the parameters under control are:

- National latency
- International latency
- Loss of packages
- Service availability
- Bandwidth reserve in order to manage traffic peaks without QoS degradation.

Different kinds of links are available in order to cover a wide range of customer requirements.

The last category offer the so called Best Effort traffic, generally devised for Internet access as well as batch and offline transmission data like e-mail. No guarantees are provided for availability, latency, jitter and loss of packages. This category can be encompassed in traditional IP traffic; consequently, data packages are immediately discarded in case of network gridlock.

Better traffic qualities assure a reasonable good QoS for corporate as well as high priority traffic like application connections, SNA traffic, database queries and so on.

Finally, the higher ranked links are designed for high priority QoS links devoted for voice and video transmissions. This kind of traffic admits very low latency and jitter. Consequently this transmission has the highest priority. These capabilities are possible due to the network intelligence offered by MPLS technology which allows carriers to provide customer bandwidth while QoS is properly assured.

Other advantages are the rather large PoP all over the country as well as the commercial agreements with international carriers in order to deploy international corporate networks. This kind of technologies were offered initially for large carriers to offer national and last mile MPLS connections to corporations that own MPLS networks all over the world. Currently, the discussed advantages like VPN's and QoS offered by MPLS permit national carriers to add value to classical IP links.

6. CONCLUSION.

MPLS is the last step in the evolution of multilevel commutation technologies (or IP commutation). The basic idea of partitioning the data batch (through and algorithm of label exchange) from the standard IP routing procedures has approached levels 3 and 2 with the resulting benefit of enhanced performance and architectural flexibility.

On the other hand, the fact that MPLS can work on any distribution technology –not only ATM infrastructures– will enable significantly the migration to the next generation of optic internet, bridging the distance between IP network level and fiber.

MPLS grants IP providers the opportunity to offer new services which are attainable with the current IP routing techniques (most commonly limited to direct by destination address). Besides enabling IP traffic engineering, MPLS allows to maintain service classes and can withstand with great efficiency VPNs creation. Consequently, MPLS is a great promise and hope to sustain the current rate of internet growth.

7. ACKNOWLEDGEMENTS.

The financial support provided by Agencia Nacional para la Promoción Científica y Tecnológica and CITEFA (Project PICTO 11-0821, Préstamo BID 1726 OC-AR) is gratefully acknowledged.

REFERENCES.

[1] Blake, S.; Black, D.; Carlson, M.; Davies, E.; Wang, Z.; Weiss, W. An Architecture for Differentiated Services. RFC 2475. 1998.
[2] Braden Ed. A Resource Reservation Protocol (RSVP). RFC 2205. 1997.
[3] Stallings. W. Data and Computers Communications - 8th Edition. Prentice Hall. 2007.
[4] Rosen E. Multiprotocol Label Switching Architecture. RFC 3031. January 2001.
[5] Rosen E. MPLS Label Stack Encoding, RFC 3032, January 2001.
[6] Pepelnjak, I., Guichard, J. MPLS and VPN Architectures. Cisco Press. 2001.
[7] Black, U. MPLS and Label Switching Networks. Prentice Hall. 2001.

[8] Stallings, W. MPLS. The Internet Protocol Journal, 2-14, Vol. 4, N. 3, 2001. Available at www.cisco.com/ijp.
[9] Canalis, M.S. MPLS “Multiprotocol Label Switching. Una tecnología de backbone para el siglo XXI. Jornadas de Informática del Noroeste Argentino. JINEA 2002. UNNE. 2002.
[10] Xiao X. Traffic Engineering with MPLS in the Internet Network, 28-33, Vol. 14, N. 2, Mar/Apr 2000. IEEE, Doi 10.1109/65.826369.
[11] Awduche D. Requirements for Traffic Engineering over MPLS. RFC 2702. September 1999.
[12] Alvarez, S. MPLS TE Technology Overview chapter 2 in QoS for IP/MPLS Networks. Cisco Press. 2006.
[13] Spurgeon, C. E. Ethernet, the Definitive Guide. O'Reilly. 2000.
[14] Seifert, R. The Switch Book. John Wiley. 2000.
[15] Carrara E., Mon J., Fusaro R. Sistemas Emergentes para la Transmisión de Datos. Redes MPLS con Calidad de Servicio. Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales - Universidad de Morón. 2007.

Análise de Protocolos de Roteamento em MANETs como Apoio a Pervasividade em Ambientes Ubíquos

Ricardo de Oliveira Schmidt, Marco Antônio Sandini Trentin
Curso de Ciência da Computação, Universidade de Passo Fundo
Passo Fundo, RS, Brasil
{60273, trentin}@upf.br

Abstract

The scenarios of Ubiquitous Computing are becoming reality in the last years. The evolution in the technologies of wireless networks is making possible the existence of these ubiquitous scenarios. Mobile ad hoc networks (MANETs) are one of the main evolutions in networks area. These networks make possible the creation of communication structures in environments, until the moment, considered impossible. This way, the objective of this work was to evaluate two MANETs routing protocols in a scenario with high mobility: DYMO and OLSR. It also presents the methodology proposed to make the simulations used in the performance evaluation of the protocols. The simulations results are presented through performance lines that make possible constructing the conclusions about the behavior of the evaluated protocols.

Keywords: pervasive environments, routing, ad hoc networks, network simulations.

Resumo

Os cenários de Computação Ubíqua estão se tornando realidade nos últimos anos. A evolução das tecnologias em redes de computadores sem fio tem possibilitado a existência de tais cenários ubíquos. Dentre as principais evoluções em redes estão as redes móveis *ad hoc* (MANETs), por propiciarem a criação de estruturas de comunicação em ambientes até então impraticáveis. Diante disso, este trabalho teve por objetivo analisar dois protocolos de roteamento para MANETs em um ambiente com alto grau de mobilidade: DYMO e OLSR. É também apresentada a metodologia proposta para a realização das simulações utilizadas na avaliação do desempenho dos protocolos. Os resultados das simulações são apresentados através de curvas de desempenho que possibilitaram a formulação das conclusões sobre o comportamento dos protocolos avaliados.

Palavras-chave: ambientes pervasivos, roteamento, redes *ad hoc*, simulações de redes.

1 INTRODUÇÃO

No início da década de 90 Mark Weiser formulou o conceito de Computação Ubíqua. Segundo Weiser, a primeira era da computação foi relacionada aos mainframes onde vários usuários estavam conectados a uma única estação de trabalho. A segunda era da computação iniciou com a popularização dos computadores e o surgimento dos computadores pessoais (PC – *Personal Computers*). Hoje estamos em uma fase de transição entre a segunda era da computação e a Computação Ubíqua, tida como terceira grande era da computação [7]. O conceito básico da Computação Ubíqua é de que vários dispositivos, comunicantes entre si, estejam conectados a um

único usuário fornecendo diversos serviços computacionais. O tamanho dos dispositivos computacionais cada vez mais reduzido e suas capacidades de processamento e armazenamento cada vez maiores são evoluções atuais que possibilitarão a criação de ambientes ubíquos. Outro fator que contribui muito com a concretização da Computação Ubíqua é a evolução das tecnologias de comunicação das redes de computadores.

As redes de computadores são essenciais na criação de um ambiente ubíquo. Como todos os dispositivos do ambiente necessitam trocar informações para trabalhar de forma autônoma, a conectividade é um fator crucial na Computação Ubíqua. A conectividade é a garantia do funcionamento do ambiente ubíquo desde a troca de informações entre os dispositivos até o reconhecimento de um novo dispositivo integrante na rede de comunicação. Um alto grau de pervasividade[1] é requerido em ambientes ubíquos. Isso devido ao fato de que um ambiente ubíquo deve ser pensando como uma rede de comunicação com diversos dispositivos wireless e, muitas vezes, móveis.

Muitos cuidados devem ser tomados com relação à rede de comunicação devido à complexidade dos ambientes ubíquos. Isso porque a rede de comunicação pode ser local ou não, através de cabos ou wireless, mas sempre levando em consideração a pertinência da conexão. A rede deve trabalhar de forma autônoma onde cada novo dispositivo deve integrar-se a rede tornando pública a sua presença e aderindo as exigências da rede.

Como já mencionado, a utilização de tecnologias wireless em redes de computadores está crescendo. As estruturas sem fio possibilitam uma maior abrangência de aplicações para as redes de computadores. Cenários com características que dificultam ou até mesmo impossibilitam a implementação de redes estruturadas, podem ser facilmente resolvidos através da comunicação wireless. Mais especificamente, em cenários militares, a utilização de redes wireless é indispensável. Hoje em dia, o campo de batalha faz uso extensivo de comunicações sem fio. Em uma operação militar, a comunicação é fundamental uma vez que essa pode acarretar o sucesso ou fracasso da missão [5]. Sendo assim, essas estruturas devem apresentar uma boa conectividade e segurança na distribuição e entrega das informações.

As redes wireless são classificadas em duas categorias: redes wireless infra-estruturadas e redes wireless não infra-estruturadas. A saber, redes wireless infra-estruturadas correspondem às redes de comunicações que fornecem um grau de mobilidade aos seus usuários através de uma Estação de Suporte à Mobilidade (AP – *Access Point*) [1]. Essas redes são utilizadas quando já existe alguma outra estrutura de comunicação fixa e há a necessidade de prover mobilidade aos usuários, mesmo que essa mobilidade seja limitada à área de cobertura do AP. Por outro lado, as redes wireless não infra-estruturadas, também conhecidas por redes *ad hoc*, não necessitam uma conexão com uma outra estrutura para existir. As redes *ad hoc* são redes dinâmicas, algumas vezes temporárias, onde os nodos integrantes da rede de comunicação são responsáveis pelo gerenciamento e manutenção da mesma. Ou seja, uma rede *ad hoc* não possui um controle centralizado responsável pela manutenção da rede. Uma rede *ad hoc* é chamada de rede não infra-estruturada devido ao dinamismo em sua topologia [4]. Ainda com relação às redes *ad hoc*, essas podem ser móveis. As redes móveis *ad hoc* são apresentadas a seguir.

2 MANETs

Uma rede móvel *ad hoc* (MANET – *Mobile Ad hoc Network*) corresponde a uma rede *ad hoc* com nodos móveis. Ou seja, além de não possuir um controle centralizado, em uma MANET os nodos, que também são responsáveis por gerenciar a rede de comunicação, movimentam-se com diferentes sentidos e velocidades (ver Figura 1). Essas estruturas são bastante complexas, pois os nodos

[1] Computação pervasiva: responsável pela adaptação dinâmica de um dispositivo a um determinado ambiente.

móveis precisam gerenciar a rede de comunicação, ou seja, tornar-se comunicável aos outros integrantes da rede. Uma MANET compreende um sistema autônomo de comunicação entre dispositivos móveis, que opera isoladamente podendo possuir uma ligação com uma rede fixa [2].

Figura 1: Rede *ad hoc*.

Dentre as aplicações de uma MANET, pode-se citar a comunicação em um cenário de operação militar, onde os integrantes dessa operação necessitam comunicar-se diretamente com outro integrante independente da posição ou existência de um controle centralizado na rede de comunicação (ver Figura 2).

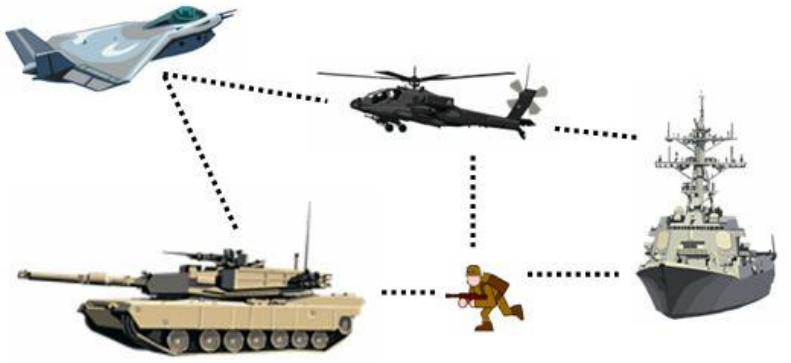


Figura 2: Cenário de comunicação militar.

Devido às suas complexas características e aplicações, as MANETs exigem eficientes protocolos de roteamento capazes de organizar e gerenciar dinamicamente a estrutura de comunicação. Dentre as principais características de uma MANET, vale-se [2]:

- Topologia dinâmica: os nodos integrantes do sistema são livres para se moverem arbitrariamente;
- Largura de banda: apesar dos notáveis avanços em tecnologias wireless, ainda é significativa a baixa capacidade de transmissão de dados. O real *throughput* de uma rede wireless é influenciado pelo efeito de múltiplos acessos, interferências, entre outros problemas encontrados durante uma transmissão;
- Operação com baixo consumo de energia: devido a uma MANET ser formada por dispositivos móveis, o consumo de energia deve ser controlado. Isso porque muitas vezes esses dispositivos móveis possuem fonte de energia não durável como, por exemplo, baterias;
- Segurança física limitada: uma rede wireless é muito mais sucessível a ataques do que uma rede fixa. Ataques como Negação de Serviço (*Denial of Service*) são mais comuns em redes sem fio.

2.1 Roteamento em MANETs

Roteamento, em redes de computadores, é a parte do software da camada de rede responsável pela decisão sobre a linha de saída a ser utilizada na transmissão do pacote de entrada [6]. Em uma

MANET, devido à ausência de um controle centralizado, há a necessidade de que os nodos integrantes da rede de comunicação também sejam capazes de realizar as tarefas de roteamento.
Um protocolo de roteamento, atuando nos nodos integrantes de uma rede móvel *ad hoc*, deve ser capaz de estabelecer dinamicamente rotas (de saltos) através da rede, bem como a adaptação dessas rotas, ou busca por novas, mesmo que esses nodos comunicantes se encontrem em movimento. Um método natural para tentar prover o roteamento em uma rede *ad hoc* é simplesmente tornar cada nodo um roteador e buscar um protocolo comum entre eles [3].
O design de um protocolo de roteamento para redes *ad hoc* ou, principalmente, para MANETs, possui características bastante complexas. A qualidade do enlace de comunicação, perda de rota (ou canal) de propagação, interferências, custo de energia e mudanças dinâmicas de topologia são fatores bastante relevantes ao tratar-se da comunicação. Novamente tomando como exemplo a utilização militar das MANETs, a preservação da segurança, latência, confiabilidade, rápida recuperação em caso de falhas, entre outros fatores são considerações significativas. Isso porque redes militares são desenvolvidas com o objetivo de manter uma baixa probabilidade de interceptação e/ou detecção.
Hoje é muito comum a utilização de simulações para avaliar o desempenho e o comportamento de uma tecnologia ou estrutura em MANETs. Isso devido às, já apresentadas, complexidades desses ambientes. As simulações possuem o objetivo de auxiliar na análise de uma tecnologia ou estrutura e levantar resultados que validem ou não sua aplicação em um ambiente real. No capítulo seguinte são apresentados a metodologia na realização de simulações sobre dois atuais protocolos de roteamento em MANETs. Através dessas simulações foram formulados resultados significativos sobre os protocolos avaliados com relação a sua utilização em cenários de operações militares.

4 SIMULAÇÕES

Simulações, não somente de redes de computadores, têm o objetivo de avaliar determinada estrutura ou tecnologia. As informações conseguidas através dessas simulações devem ser confiáveis ao ponto de interferirem no processo de decisão da aplicação da estrutura ou tecnologia avaliada em ambiente real. Mais especificamente, simulações de redes móveis *ad hoc* têm o objetivo de simular ambientes geralmente impraticáveis por questões físicas ou por custos de implementação. MANETs ainda são estruturas complexas e difíceis de implementar. Portanto, simulações são importantes à medida que possam fornecer informações confiáveis sobre a estrutura de comunicação e auxiliar em tomadas de decisão durante o desenvolvimento de uma determinada tecnologia ou implantação da mesma.
Em um cenário militar a rede de comunicação é de fundamental importância uma vez que irá garantir a comunicação necessária durante uma determinada operação. A escolha pela simulação de um cenário militar deu-se devido a esse indispensável cuidado com a rede de comunicação que irá proporcionar a conectividade em um ambiente hostil e imprevisível. A metodologia utilizada nas simulações e o cenário simulado estão descritos a seguir.

4.1 Metodologia

A escolha da correta metodologia para a realização de simulações em MANETs é muito importante pois a validade e confiabilidade dos resultados dependem diretamente da mesma. Para a realização das simulações em um cenário militar foi proposta uma estrutura de simulação com 4 ferramentas principais, sendo:

- ScenGen (*Scenario Generator*): utilizado para a geração de *scripts* de movimentação dos nodos integrantes da rede de comunicação;

- Cbrgen.tcl: utilizado para a geração de *scripts* com todas as conexões para a transmissão de pacotes de dados entre os nodos integrantes da rede;
- NS-2 (*Network Simulator 2*): simulador de redes que reúne as informações dos *scripts* gerados pelo ScenGen e pelo Cbrgen.tcl com outras especificações da rede de comunicação, realiza a simulação e gera os arquivos resultantes com todas as informações significativas da rede durante o tempo simulado;
- *Awk*: linguagem de programação que possibilita a formulação de *scripts* para a extração de informações de arquivos com estrutura definida (informações em colunas). A utilização do *awk* deveu-se ao fato de que os arquivos resultantes das simulações no NS-2 são muito extensos, possuindo milhares de linhas de informações. Essas linhas seguem uma padronização na apresentação do conteúdo desses arquivos. Os *scripts* desenvolvidos em *awk* tiveram o objetivo de extrair informações significativas desses arquivos de resultado.

Através da Figura 3 podem ser visualizados todos os *scripts* necessários e a seqüência da utilização de cada uma das ferramentas citadas acima.

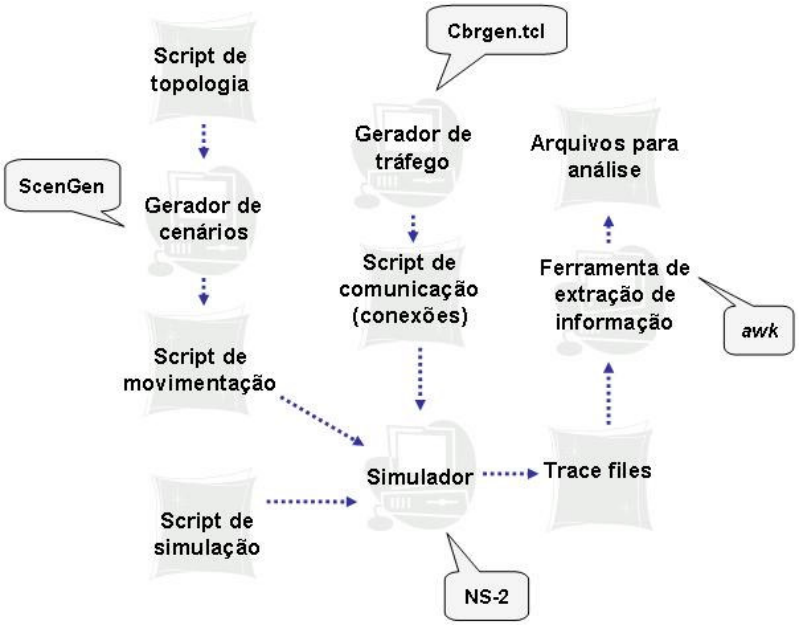


Figura 3: Estrutura de simulação.

4.2 Cenário proposto

O cenário proposto para a realização das simulações consiste na operação militar de assalto e tomada de posição inimiga. Nessa situação, um ou mais grupos de soldados buscam alcançar e tomar uma posição até o momento sob posse do inimigo. Nesse tipo de operação não é esperada uma forte reação por parte do inimigo devido ao elemento surpresa do ataque. O cenário simulado faz parte do trabalho realizado em [5].

Como é apresentado mais adiante, na Figura 4, o cenário simulado é constituído por 35 unidades. Todas as unidades possuem um comunicador capaz de atuar em uma rede *ad hoc*. Essas unidades partem de um ponto inicial no cenário de simulação e buscam alcançar um ponto em comum que é a posição inimiga, ou seja, o que deve ser a posição final das unidades ao término da simulação. As 35 unidades estão divididas e organizadas no cenário de simulação da seguinte forma: 1 central de

comando; 2 observadores avançados; e 8 grupos de 4 integrantes cada. A área total da simulação é de 1000m X 2000m.
Ainda é importante saber que no cenário simulado foi estabelecido que a central de comando (carro de combate) é o ponto de referência para todos os outros integrantes da rede. Isso porque a mesma dita a movimentação do grupo através de informações pré-estabelecidas e das obtidas durante a execução da operação. Sendo assim, as conexões simuladas representam a comunicação dos grupos e observadores com a central de comando. Para essas comunicações foi estabelecido, de forma aleatória, um nodo responsável por cada grupo. Isso totalizou 10 conexões chegando a central de comando (uma por cada um dos grupos e uma por cada observador) e 10 conexões saindo da central de comando como resposta. A comunicação entre os próprios integrantes dos grupos não foi considerada para a análise da rede, pois devido à proximidade entre os nodos dos grupos, todos estavam dentro da área possível de comunicação de seus companheiros. Isso proporciona atingir facilmente os 100% de comunicação efetiva, ou seja, toda a comunicação necessária é conseguida.

4.2.1 Modelo de mobilidade proposto

Como mencionado no item 4.1, foi utilizada a ferramenta ScenGen para auxiliar na geração da movimentação dos nodos móveis da rede simulada. Para isso há a necessidade de um modelo de mobilidade responsável por ditar as regras que essa movimentação deve obedecer. No caso do cenário de operação militar proposto, os nodos móveis devem seguir uma linha de movimentação de forma a atuarem em conjunto e alcançarem o ponto final em comum.
Para modelar a movimentação de todos os nodos integrantes da rede de comunicação foi utilizado o modelo de mobilidade *Mixed Waypoint*, desenvolvido por [5]. Esse modelo de mobilidade é uma mescla dos modelos *Random Waypoint* e RPGM. O modelo *Mixed Waypoint* é escrito em C++ e foi integrado à ferramenta ScenGen para que pudesse ser utilizado de forma a tornar a movimentação das unidades mais próxima de um ambiente real.
A movimentação dos nodos consistiu em três movimentações principais para cada grupo integrante da operação. Ou seja, partindo do ponto inicial, os grupos de nodos (considerando a central de comando como um grupo e os dois observadores também como um grupo) se movimentavam até uma segunda área. Após um tempo de pausa, os grupos se movimentavam novamente em direção ao ponto final até um terceiro ponto. E novamente após um tempo de pausa, os grupos se movimentavam até o ponto final em comum a todos. Essa movimentação, bem como a estrutura completa do cenário simulado, pode ser observada na Figura 4.

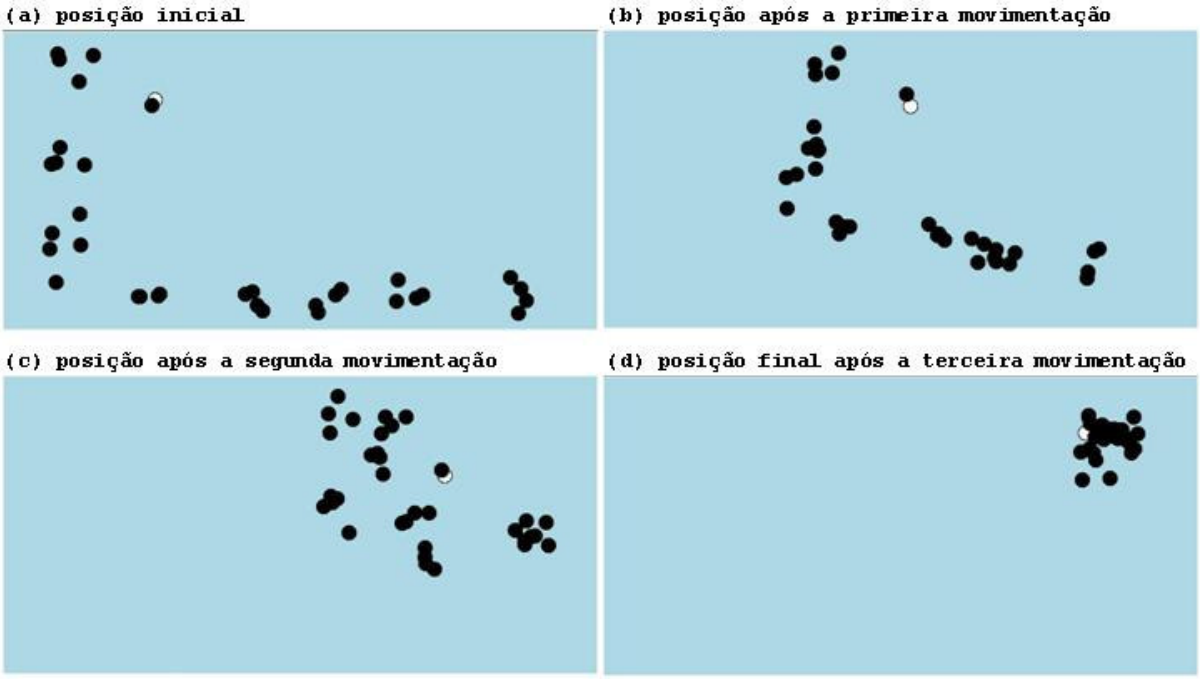


Figura 4: Movimentação no cenário simulado gerada através do ScenGen e visualizada através do Ad-hockey.

5. RESULTADOS E CONCLUSÕES

As simulações realizadas tiveram o objetivo de analisar o desempenho dos protocolos OLSR[2] (pró-ativo) e DYMO[3] (reativo) no cenário descrito no item 4. Foram selecionados tais protocolos devido ao fato de serem protocolos novos e ainda estarem em desenvolvimento.

Ao total foram realizadas 30 rodadas de simulação, sendo 15 para cada protocolo. Através da realização de mais de uma rodada de simulação, buscou-se uma maior proximidade a um cenário real. As movimentações principais dos grupos e as comunicações existentes entre os nodos são as mesmas em todas as rodadas de simulação. Porém, a movimentação individual de cada nodo dentro de seu respectivo grupo é alterada. Essa alteração pode acarretar mudanças na forma de comunicação entre uma simulação e outra. Uma conexão existente entre dois nodos em determinada rodada de simulação, por exemplo, pode não ser possível em outra rodada devido à diferente direção tomada por um dos nodos, ou ambos, dentro de seu grupo. Sendo assim, para a formulação dos resultados finais fez-se a média dos resultados parciais individuais de cada rodada de simulação.

As métricas utilizadas para comparar o desempenho dos protocolos nas simulações realizadas são as seguintes:

- Taxa de entrega de pacotes: razão entre o número de pacotes de dados entregues no destino final e o número de pacotes de dados gerados pela aplicação na fonte;
- Atraso médio fim a fim dos pacotes de dados (*delay end-to-end*): inclui todos os possíveis atrasos causados por latência da descoberta de rotas, propagação, retransmissões da camada MAC e tempos de transferência;

[2] Código-fonte UM-OLSR. Desenvolvido por Francisco J. Ros. Masimum – Universidade de Murcia, Espanha.
[3] Código-fonte DYMOUM. Desenvolvido por Francisco J. Ros. Masimum – Universidade de Murcia, Espanha.

- Número de pacotes e bytes de roteamento: a quantidade total de pacotes e bytes de roteamento gerados por cada um dos protocolos. Essa métrica compreende a descoberta de rotas roteamento e a manutenção das mesmas;
- Sobrecarga de roteamento normalizada pelo número de pacotes e bytes de dados: razão entre o número de pacotes e bytes de roteamento transmitidos na rede e, respectivamente, a quantidade de pacotes e bytes de dados entregues aos seus respectivos destinos.

A Tabela 1 apresenta o resumo das principais características da simulação realizada.

Tabela 1: Resumo dos parâmetros da simulação.

Parâmetro	**Valor**
Largura de banda	2Mbps
Alcance de transmissão	250m
Tempo total de simulação	500s
Tempo de pausa (máx.)	5s
Número total de nodos	35
Área de simulação	2000m x 1000m
Tamanho dos pacotes de dados	512 bytes
Tipo de tráfego	CBR (*Constant Bit Rate*) – UPD
Número de conexões	20

Com relação aos gráficos apresentados na Figura 5, é importante saber que os momentos de crise na estrutura de comunicação acontecem no intervalo entre 150 e 200 segundos e no intervalo entre 400 e 450 segundos de simulação. Isso ocorre devido a segunda e a terceira movimentação em conjunto de todos os grupos da rede, como pode ser visto na Figura 4(c) e na Figura 4(d). As movimentações dos grupos acarretam uma grande perda nas informações relativas às rotas de roteamento que estavam sendo utilizadas até o momento. Também se pode observar que as linhas do gráfico tendem ao melhor valor da métrica com o passar do tempo de simulação. Isso ocorre devido à proximidade entre os nodos após as movimentações, pois os mesmos estão caminhando para um objetivo em comum. Dessa forma, no intervalo entre 450 e 500 segundos as linhas alcançam valores melhores nas métricas, pois os nodos estão muito próximos uns dos outros e toda a comunicação necessária é conseguida.

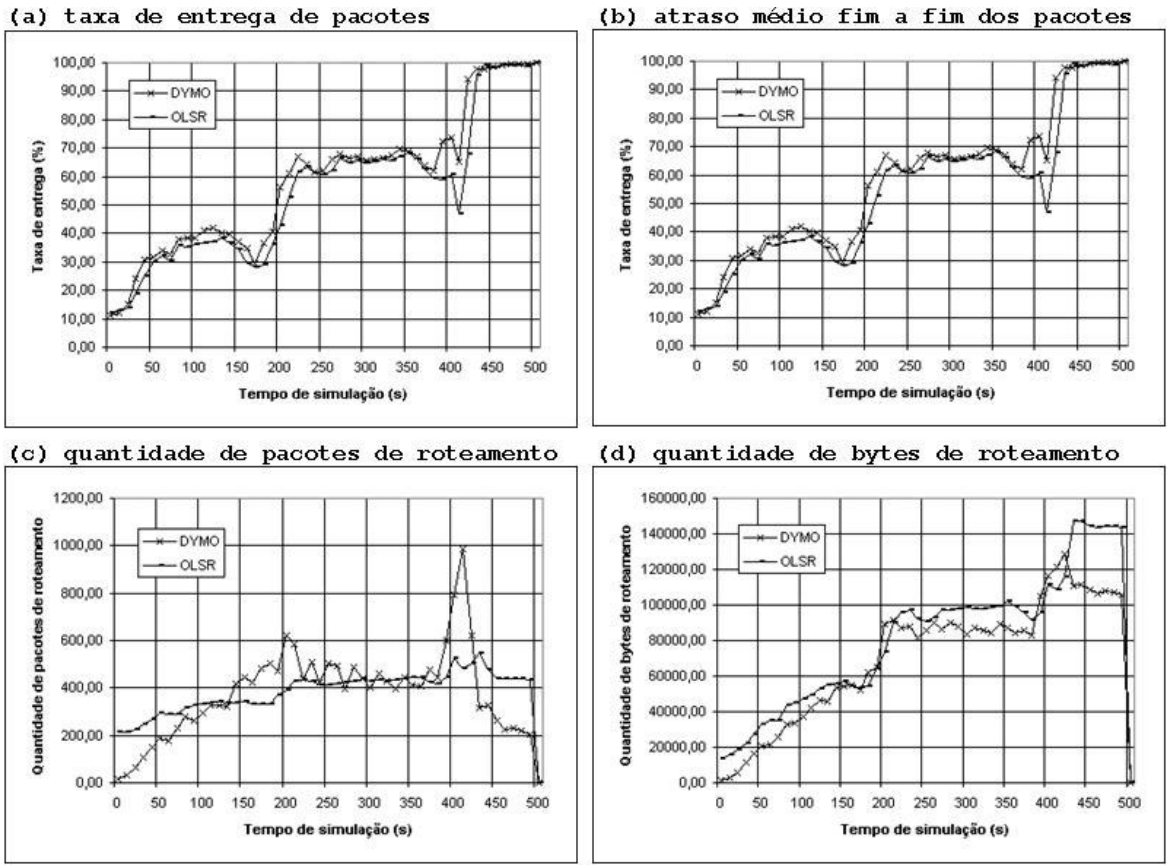


Figura 5: Gráficos com os resultados da simulação.

Nas simulações realizadas, o protocolo DYMO apresentou-se mais eficiente que o protocolo OLSR principalmente no que diz respeito à taxa de entrega dos pacotes de dados. Como pode ser observado no gráfico "a" da Figura 5, o protocolo DYMO consegue atingir melhores níveis na entrega dos pacotes de dados mais rápido que o protocolo OLSR. O DYMO também, em alguns momentos da simulação consegue atingir picos na entrega dos pacotes de dados que o protocolo OLSR não consegue como, por exemplo, no intervalo entre 200 e 250 segundos de simulação. É importante ressaltar também que o protocolo DYMO, trabalhando de forma reativa, nos momentos de crise da estrutura de comunicação, apresentou uma melhor recuperação na reconstituição da comunicação. O protocolo OLSR demorou mais tempo para conseguir restabelecer as conexões perdidas durante a movimentação dos grupos.

O gráfico "b" da Figura 5 pode apresentar uma idéia contrária com relação ao desempenho dos protocolos. Isso porque a linha que representa o atraso fim a fim dos pacotes de dados para o protocolo OLSR é muito mais constante que a do protocolo DYMO. Isso se deve a forma de atuação sob demanda (reativa) do protocolo DYMO. Como também é somada ao tempo de atraso a latência ocasionada pelos trabalhos necessários de roteamento (descoberta e manutenção de rotas), o trabalho reativo do protocolo DYMO provoca picos muito mais elevados em tempo de atraso que o protocolo OLSR que possui uma linha muito mais constante devido ao trabalho de roteamento pró-ativo. Pode-se observar ainda no gráfico "b", que nos momentos de crise na rede ambos os protocolos se comportaram de maneira semelhante. Pois a perda das rotas de comunicação ocasionou um maior atraso na entrega dos pacotes que foram gerados e enviados nos momentos de crise.

O gráfico "c" da Figura 5 comprova o comportamento mais equilibrado por parte do protocolo OLSR. O protocolo OLSR gerou pacotes de roteamento à medida que necessitava preencher ou

atualizar sua tabela de roteamento. A atualização das tabelas de roteamento era realizada a cada 2 segundos de simulação. Já o protocolo DYMO apresenta uma linha crescente nos primeiros segundos de simulação devido ao trabalho reativo. Ou seja, à medida que pacotes de dados eram gerados, o protocolo DYMO iniciava a busca por rotas, caso a mesma ainda não tivesse sido descoberta, para realizar a entrega desses pacotes. Nos momentos de crise na rede de comunicação, o protocolo DYMO gerou um número maior de pacotes de roteamento que o protocolo OLSR. Sendo assim o protocolo DYMO conseguiu restabelecer mais rapidamente as rotas necessárias para continuar o envio dos pacotes de dados. Isso explica a melhor recuperação na taxa de entrega dos pacotes de dados pelo protocolo DYMO, apresentada no gráfico "a" da Figura 5.

No gráfico "d" da Figura 5, se pode observar que a quantidade de bytes de roteamento enviada pelos protocolos durante a simulação foi bastante equilibrada. Sendo que em alguns momentos o protocolo OLSR enviou mais bytes que o protocolo DYMO e em outros momentos aconteceu o inverso. Nos momentos de crise na rede de comunicação, o protocolo OSLR gerou uma carga maior de bytes de roteamento. Mas isso não refletiu em um melhor desempenho com relação à taxa de entrega dos pacotes de dados. Ou seja, nos momentos de crise na comunicação, apesar da maior quantidade de pacotes de roteamento gerados, o protocolo DYMO conseguiu melhorar ou manter a taxa de entrega dos pacotes de dados gerando uma carga menor de bytes de roteamento. É importante salientar que a quantidade de pacotes de roteamento não interfere de forma direta no resultado relativo à quantidade de bytes de roteamento. Isso porque os pacotes de roteamento dos protocolos avaliados são diferentes, pois um trabalha de forma pró-ativa e outro de forma reativa. Sendo assim, os pacotes de roteamento dos dois protocolos levam dados diferentes em seu conteúdo interferindo diretamente no seu tamanho. O protocolo OLSR, durante todo o tempo de simulação, gerou uma carga de aproximadamente 61MB em tarefas de roteamento, cerca de 13% a mais que o protocolo DYMO que, sob as mesmas condições, gerou uma carga de aproximadamente 53MB em roteamento.

Apesar de demonstrar um trabalho de roteamento constante, por atuar de forma pró-ativa, o protocolo OLSR, no cenário avaliado, necessitou uma média de 3,89 pacotes de roteamento para que um pacote de dados conseguisse alcançar seu destino final. O protocolo DYMO para o mesmo trabalho necessitou uma média de 3,6 pacotes de roteamento. A diferença média de 0,29 pacotes de roteamento parece ser insignificante. Porém, deve-se considerar que em 500 segundos de simulação milhares de pacotes de dados são gerados e enviados, o que torna essa diferença um valor bastante expressivo. Da mesma forma que na quantidade de pacotes de roteamento, o protocolo OLSR necessitou uma maior quantidade de bytes de roteamento para a entrega dos pacotes de dados. O protocolo OLSR gerou em média 1,54 bytes de roteamento para cada byte de dados entregue ao seu destino final. O protocolo DYMO necessitou que em média 1,26 bytes de roteamento fossem gerados para que cada byte de dados chegasse ao seu destino. Nessa métrica vale a mesma conclusão com relação à carga de roteamento calculada pela quantidade de pacotes. Isso porque a diferença parece ser pequena ao se tratar de um byte de dados. Porém, ao longo de 500 segundos de simulação, milhares de pacotes de dados, com tamanho de 512 bytes, foram gerados. Isso faz com que essa pequena diferença se torne significativa ao término do tempo total de simulação.

Após a realização das 15 rodadas de simulação para cada um dos protocolos avaliados e a análise dos resultados, chegou-se a conclusão de que ambos os protocolos conseguiram realizar as tarefas de roteamento de forma a possibilitar a comunicação entre os integrantes do cenário. Porém, o protocolo DYMO apresentou um melhor desempenho sob as condições simuladas. Essa conclusão é devido ao fato de que o protocolo conseguiu estabelecer melhores taxas de entrega e de recuperação em momentos de crise na comunicação, atuando de forma reativa. Ainda é válido colocar que essas melhores taxas foram conseguidas com uma menor carga de roteamento durante todo o tempo de simulação. O protocolo OLSR apresentou um melhor equilíbrio com relação ao atraso na entrega dos pacotes de dados. Porém isso não foi suficiente para que uma melhor taxa de entrega fosse

atingida e, num cenário de operação militar com uma central de comando que dita as direções do grupo, essa métrica é de grande importância. Da mesma forma, é de grande importância a menor carga de roteamento, pois isso significa uma menor quantidade de informações trafegando na rede de comunicação. Isso acarreta uma maior discrição da rede de comunicação no campo de batalha. Ou seja, quanto menor a quantidade de dados transmitidos, menor as chances da rede de comunicação ser detectada pelo inimigo.

Com a realização desse trabalho chegou-se a conclusão de que o protocolo DYMO atende melhor as exigências de uma rede de comunicação em um cenário de operação militar com as características apresentadas no item 4. Porém, muitas outras qualidades de um protocolo ainda devem ser avaliadas para que o mesmo possa ser escolhido como tecnologia a ser utilizada em um ambiente real. Outros fatores ainda não avaliados no trabalho realizado, como a segurança de dados, também são determinantes para o correto funcionamento da rede de comunicação e, consecutivamente, o sucesso ou fracasso da missão. Portanto esse trabalho veio contribuir à medida que foi possível avaliar o desempenho dos protocolos OLSR e DYMO de forma mais abrangente e não específica sobre uma única característica. Mas com esse trabalho já é possível alegar a superioridade do protocolo DYMO no cenário avaliado.

REFERÊNCIAS

[1] Corrêa, U. et al. Redes Locais Sem Fio: Conceitos e Aplicações. *In: IV Escola Regional de Redes de Computadores*. Passo Fundo, Brazil. 2006.

[2] Corson, S. and Macker, J. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. 1999.

[3] Johnson, D. Routing in Ad hoc Networks of Mobile Hosts. *In: IEEE Workshop on Mobile Computing Systems and Applications.* 1994.

[4] Lundberg, D. Ad hoc Protocol Evaluation and Experiences of Real World Ad hoc Networking. Departament of Technology, Uppsala University. Uppsala, Sweden. 2002.

[5] Pereira, I.C.M. Análise do Roteamento em Redes Móveis Ad hoc em Cenários de Operações Militares. Universidade Federal do Rio de Janeiro. Rio de Janeiro, Brazil. 2004.

[6] Tanenbaum, A.S. Redes de Computadores. Editora Campus. Rio de Janeiro, Brazil. 1997.

[7] Weiser, M. Ubiquitous Computing. http://sandbox.xerox.com/unicomp/. 1996.

Análisis del Protocolo MAC de 802.15.3 bajo Esquemas de Prioridades en Tiempo Real

Guillermo H. Reggiani
Dpto. Electrónica, UTN – FRBB
Bahía Blanca, Bs. As. 8000, Argentina
ghreggiani@frbb.utn.edu.ar

Omar R. Alimenti
Dpto. Electrónica/DIEC, UTN – FRBB/UNS
Bahía Blanca, Bs. As. 8000, Argentina
iealimen@uns.edu.ar

Ricardo L. Cayssials
DIEC, Universidad Nacional del Sur (UNS)
Bahía Blanca, Bs. As. 8000, Argentina
iecayss@criba.edu.ar

and

Ferro C. Edgardo
DIEC, Universidad Nacional del Sur (UNS)
Bahía Blanca, Bs. As. 8000, Argentina
eferro@uns.edu.ar

Resumen

La calidad de servicio (QoS) con requerimientos de tiempo real aplicado a diversas implementaciones como multimedia y manejo de sensores inteligentes, sobre redes inalámbricas ("wireless") de área personal (WPANs-802.15.3), depende de muchos factores como el algoritmo de asignación de tiempo del canal y el tamaño de la supertrama ("superframe") entre otros. En este trabajo se propone una metodología, basada en simulaciones, que permite estimar el tamaño de supertrama para brindar el mayor rendimiento al sistema en cada uno de los dos casos de asignación de prioridades estudiados: EDF y SRPT, sin modificar la estructura de la trama.

Palabras Claves: Supertrama, Wpan, Redes Inalámbricas, Diagramación, "Beacon", Prioridades.

Abstract

The quality of service (QoS) with requirements of real time applied to diverse implementations like multimedia and handling of intelligent sensors, over wireless nets of personal area (WPANs-802.15.3), depends on many factors like the channel time allocation algorithm and the size of the superframe among others. In this paper we propose a methodology, based on simulations that allow estimating the superframe size to offer the biggest performance to the system for each one of the two studied cases of assignment: EDF and SRPT, without modifying the frame structure.

Keywords: Superframe, Wpan, Wireless Networks, Scheduling, Beacon, Priorities.

1 Introducción

El Standard 802.15.3 [4] especifica los requerimientos en la capa física (PHY) y el control de acceso a medio (MAC) en redes inalámbricas ("*wireless*") de área personal (WPAN), que fueran capaces de transmitir datos a alta tasa de transferencia, de manera eficiente, con baja complejidad, bajo costo y mínimo consumo.
El Standard trabaja dentro de un sistema de comunicación inalámbrica de datos, denominado "*piconet*", el cual permite que un número independiente de dispositivos (DEVs) se comuniquen unos con otros. El control de acceso al medio de un conjunto de DEVs es gobernado por un coordinador de la piconet (denominado PNC) que permite compartir los recursos inalámbricos de la red.
El PNC también provee el temporizado básico para la WPAN y administra los requerimientos de calidad de servicio (QoS). El temporizado en este estándar se basa en una estructura denominada "supertrama". La supertrama está compuesta por tres partes: trama de señalización ("*beacon"*), período de contención de acceso (*CAP*) y periodo de asignación de canal de tiempo (*CTAP*). La figura 1 muestra un esquema de la supertrama.

Figura 1: Supertrama

El "*beacon*" sirve para fijar las asignaciones de tiempo y para comunicar la información de administración dentro de la piconet. Los DEVs utilizan el CAP para informar los requerimientos de uso del canal (el CAP es opcional). El CTAP esta compuesto por asignaciones de tiempo de uso del canal (CTAs). Los CTAs se emplean para enviar comandos, cadenas isocronicas y conexiones de datos asincrónicos. La asignación de las ranuras de tiempo es uno de los puntos esenciales para un buen aprovechamiento del uso del canal. En este trabajo se analizan las asignaciones de los CTAs para diferentes tamaños de supertrama, utilizando las técnicas de disciplinas de prioridades: SRPT ("Short Remaining Processing Time") y EDF ("Earliest Deadline Firts").
Además dentro de la estructura de la supertrama están los "*administradores CTA"* (MCTAs) que se utilizan para enviar pedidos de asignación de canal desde los DEVs al PNC entre otros. Los MCTA son asignados por el PNC y utilizan un mecanismo de acceso al medio denominado "*Slotted Aloha*".
En este trabajo se utilizarán los MCTA para solicitar los requerimientos de canal en lugar del CAP.
La capa PHY soporta 5 tasas de transferencia 11, 22, 33, 44 y 55 Mbps.
El trabajo se organiza como sigue: Sección 2 provee una descripción de trabajos relacionados. En la Sección 4 se analiza el estándar para cierto tipo de tráfico con distintos tipos de diagramación. En la Sección 5 se muestran los resultados de las evaluaciones efectuadas. En la Sección 5 aparecen las conclusiones y en la Sección 6, las referencias.

2 Trabajos relacionados

Esta sección describe algunos trabajos relacionados con nuestro tema propuesto.
En [1] se presenta un algoritmo dinámico de asignación de canal de tiempo que provee garantías en el retardo de tráfico multimedial como cadenas MPEG sobre redes IEEE 802.15.3 WPANs. Este algoritmo permite a los DEVs enviar al PNC, los requerimientos de tiempo de uso del canal al final de la supertrama, usando la estructura de mini-paquetes.

La duración del tiempo de uso del canal varía dinámicamente dependiendo del tipo de trama MPEG, de la carga del tráfico, límite de retardo de la trama, etc.
El algoritmo de asignación dinámica de tiempo de uso del canal es llamado FACTA (Feedback-assisted channel time allocation) y los mini-paquetes son llamados FCTAs (Feedback control time allocation). La ubicación de los FCTAs dentro de la supertrama está en el final de la supertrama para que los requerimientos que aparecen en la supertrama *n* puedan ser atendidos en la supertrama (*n+1*). Con esto se logra mejorar el rendimiento debido al incremento de las oportunidades de asignación de tiempo de uso del canal.
Aquí también se plantea cuan a menudo se asignan los FCTAs a los DEVs, dado que si estos no tienen nada para transmitir se estaría mal gastando el tiempo de uso del canal. Para esto utiliza un parámetro llamado NFTi (Next Feedback Time) el cual indica la instancia en el cual el PNC debe asignar un FCTA al DEVi.
En [2] se mejora el algoritmo de diagramación con señalización de estado de flujo y decisión de bloque elegible. La señalización de estado de flujo se realiza usando paquetes de control (CTRL). Estos son enviados por los nodos al PNC solamente cuando es necesario actualizar la información interna de estado de flujo.
La información que transporta el paquete CTRL es:

- Tiempo de Vida (Lifetime): es el tiempo de vida residual del primer bloque
- Número de paquetes: números de paquetes del primer bloque
- Tamaño de cola: tamaño total de la cola del flujo.

La asignación de la ranura de tiempo para paquetes de CTRL se realiza clasificando el flujo en diferentes estados:

Estado ciego
Estado inicial
Estado medio

Dependiendo en que estado este el PNC otorgará una ranura de tiempo dentro de la supertrama para que el DEV correspondiente envíe sus requerimientos. Hasta el momento se consideraba un bloque por supertrama. Puede ocurrir que durante una supertrama se generan muchos bloques del mismo flujo. Para prevenir esto se plantea la sumatoria de los bloques.
También se ha tomado hasta el momento que los bloques se generaban después del vencimiento del previo. Pero puede ocurrir que estos tengan una alta tasa de arribo. Para esto se plantea el algoritmo decisión de bloque elegible (BTED).
En [3] se propone un método de diagramación llamado Trafico Máximo (MT). El algoritmo MT garantiza que el total de datos transmitidos es máximo en cualquier punto del tiempo y las ranuras de tiempo necesarios para la transmisión son mínimas. Los principales aportes de este trabajo son la determinación del límite superior de las ranuras de tiempo requeridas para una red de tráfico saturado y el análisis cuantitativo el cual es adoptado para determinar la longitud del CAP. Este último se logra a través de un modelo probabilístico.
En [5] se analiza un nuevo algoritmo de formación de supertramas que provee ahorro de energía, tratamiento apropiado de trafico CBR, gran utilización del canal, mejor tolerancia a errores y estabilidad. Se diseña un algoritmo de formación gradual de supertramas. La idea principal es que éste se adapte dinámicamente a la carga de la red intentando optimizar la utilización del canal. Para optimizar el tamaño de la supertrama el algoritmo necesita alguna información acerca de la carga futura de la red. Por la tanto las técnicas de realimentación son usadas para obtener información acerca del estado interno de los nodos. Este algoritmo gradual intenta superar los problemas de ranuras de tiempos libres y mal gastadas modificando el tamaño de la supertrama.

En [6] se muestra un algoritmo de formación de supertramas jerárquico, similar a la propuesto en [5]. En éste se combinan las ventajas de los algoritmos de formación de supertramas estáticos y dinámicos. Se plantea el uso de dos tipos de supertramas: una supertrama normal con su "beacon" normal y una mini-supertrama con su mini-"beacon". Se obtuvo un sistema con alto rendimiento, tolerancia al error y características de ahorro de energía.

3 Análisis de la Red con SRTP, EDF y RMS

El análisis para la asignación del tiempo de uso del canal a nivel del MAC, en el estándar IEEE 802.15.3 (WPANs), para un ambiente en tiempo real, será el enfoque principal de este trabajo. Evaluaremos el comportamiento del estándar bajo las disciplinas de prioridades SRTP "menor tiempo remanente de proceso", EDF "menor tiempo al vencimiento" y RMS "períodos monotónicos crecientes".
En este trabajo se tratará con la norma original, excepto que utilizaremos los MCTAs en vez del CAP (el uso del CAP es opcional) para tomar los requerimientos de tiempo de uso del canal de los distintos dispositivos. La figura 2 muestra la estructura de nuestra supertrama:

B	"Channel Time Allocation Period"								
	MCTA 1	MCTA 2	****	MCTA n	CTA 1	CTA 2	CTA 3	****	CTA n

Figura 2: supertrama utilizada para este trabajo

donde B ("beacon") es la trama de señalización de la supertrama que contendrá principalmente los valores de inicio y fin de cada ranura de tiempo que se le asigna al DEV, longitud en tiempo de la supertrama y todo lo relacionado a la calidad de servicio. La cantidad de nodos está representada por n.
El PNC asignará un tiempo de uso del canal para cada uno de los MCTA correspondiente a cada uno de los DEVs. Estos enviarán en los MCTA su pedido de tiempo de uso del canal de acuerdo a su tráfico de datos.
El flujo de datos a transmitir (tarea de aquí en adelante) de cada DEV estará parametrizado por tres valores: duración de la tarea (C), período de la tarea (T) y vencimiento de la tarea (D). El C es tiempo de utilización del canal que requiere un DEV para enviar sus datos por la red, T es el período mínimo de generación de datos en el DEV y D es el plazo máximo de tiempo en el cual se debe transmitir los datos del DEV, para cumplir con las condiciones de calidad del sistema.
En nuestro análisis tomaremos diversos flujos de datos con distintos parámetros para evaluar el rendimiento de la red utilizando los tres algoritmos de diagramación con el fin de poder apreciar cuál de ellos es más apropiado en ciertas circunstancias como así también que tamaño de supertrama es más acorde.
El tamaño máximo de supertrama según la norma es 65536 µseg, pero como veremos más adelante se usaran tamaños de hasta 15000 µseg para realizar las diversas ensayos. Parecería intuitivo pensar que cuanto mayor es la supertrama se mejora el rendimiento, pero eso dependerá de varios factores como la cantidad de DEVs, los parámetros de la tarea, algoritmos de diagramación entre otros. Una mala elección de estos valores podría originar un mal aprovechamiento del canal o que muchas tareas pierdan su vencimiento.
La disciplina SRTP ejecuta primero la tarea de menor duración (menor C), EDF ejecuta las tareas de que tengan menor tiempo al vencimiento (menor D) y RMS aquellas de menor Periodo (menor T).

La asignación de los períodos MCTA y CTA ("Channel Time Allocation Period") se realiza por TDMA (acceso múltiple por división de tiempo), técnica que permite compartir un canal de comunicaciones, libre de colisiones. Dentro de la supertrama hay periodos de tiempos importantes a tener en cuenta, que permiten respetar un tiempo suficiente entre transmisiones para asegurar las comunicaciones se realicen en una forma adecuada, como por ejemplo:
BIFS: espacio entre tramas de "backoff"
RIFS: espacio entre tramas de retransmisión
SIFS: espacio entre tramas más corto
MIFS: espacio entre tramas mínimo
Estos tiempos están incluidos en los parámetros temporales de las tareas. La relación del tiempo total de la supertrama respecto del "Beacon" y los MCTAs estará directamente vinculado con la cantidad de DEVs, ya que éstos utilizan dichos campos para realizar sus requerimientos. La influencia de la duración de los "Beacon" y los MCTAs será mayor cuanto menor sea la duración de la supertrama.

4 Evaluación de Rendimiento

La evaluación de la capa MAC de una red "WPAN" fue realizada mediante la simulación de la asignación de tiempos de uso del canal de comunicaciones del estándar 802.15.3 bajo las disciplinas de prioridades EDF, SRTP y RMS. En la simulación se consideran distintas redes "piconet", variando la cantidad de DEVs, el tamaño de supertrama y distintos tipos de tráficos (C, T y D). La simulación considera los tres tipos de disciplinas de prioridades sobre la base del estándar a una tasa de transferencia de 55 Mbps. Los parámetros del sistema (C, T y D) de la primer simulación se muestran en la tabla 1, donde el tamaño de supertrama se varió de 5000 µseg a 15000 µseg en saltos de 1000 µseg y la cantidad de nodos se variaron de 2 a 10 nodos. El tiempo de simulación para cada caso es de 100000 µseg y solo se simuló EDF dado que por el patrón de tráfico de la tabla 1, las otras disciplinas de prioridades, SRTP y RMS, darían los mismos resultados. Las principales medidas de evaluación realizadas fueron el tiempo de respuesta y el número de tareas vencidas, tal como se muestra en las figuras 3 y 4.

Nodo	C [µseg]	T [µseg]	D [µseg]
2	3000	33000	33000
3	3000	33000	33000
⋮	⋮	⋮	⋮
10	3000	33000	33000

Tabla 1

El tiempo de respuesta fue tomado como el promedio de la diferencia entre los tiempos de generación de los datos en los nodos fuente y los tiempos de finalización de la transmisión de los datos en la red, incluyendo el tiempo de transmisión por la misma. En la figura 3 se puede apreciar que los tiempos de respuesta varían entre 12000 y 25000 µseg., aumentando a medida que es mayor el número de nodos y el tamaño de supertrama.
Un menor valor de supertrama logra mejor tiempo de respuesta. Note que para una red con dos nodos el crecimiento del tiempo de respuesta es mayor, entre una supertrama de 5000 µseg y una

supertrama de 15000 µseg, que para una red compuesta por 10 nodos. Además en el instante de simulación donde el tamaño de supertrama es de 11000 µseg, se produce una mejora del tiempo de respuesta debido a que ese valor es múltiplo de los periodos de arribo de información a los nodos (T). Esto significa que al momento de arribo de los nuevos pedidos se producirá el inicio de una nueva supertrama en donde los DEVs pedirán sus espacios de tiempos para transmitir y éstos serán asignados en la siguiente supertrama; en cualquier otro instante de tiempo los DEVs tendrán que esperar un tiempo mayor a una supertrama para poder transmitir. La figura 4 muestra el número de tareas vencidas que responde de la misma forma que el tiempo de respuesta, a menor tamaño de supertrama menor el número de tareas que pierden su vencimiento. En el punto de simulación donde la supertrama es 11000 µseg se observa una mejora (disminución de paquetes perdidos) debidos a los motivos explicados anteriormente.

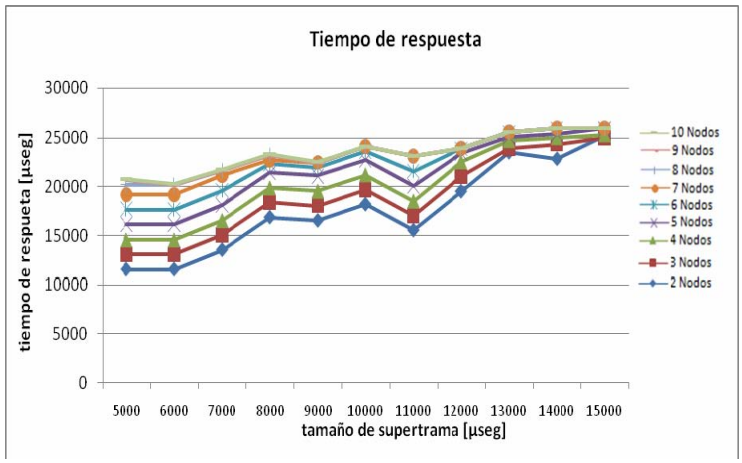


Figura 3: tiempo de respuesta

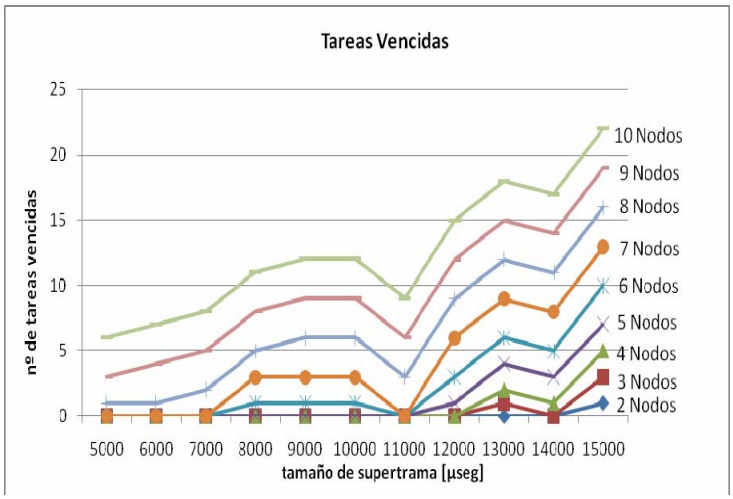


Figura 4: tareas vencidas

El segundo escenario de simulación se muestra en la tabla 2, donde se aprecia que los valores de la duración de los datos (C) y los tiempos de arribo (T) entre los distintos nodos son diferentes. En este escenario se tomó una “piconet” fija compuesta por 5 nodos y se simuló para distintos tamaños

de supertrama, variando la misma de 3000 µseg a 8000 µseg en pasos de 1000 µseg, manteniendo en 100000 µseg el tiempo de simulación. La figura 5 y 6 muestran el tiempo de respuesta y el número de tareas vencidas para este escenario.

Nodo	C [µseg]	T[µseg]	D[µseg]
1	3000	12000	12000
2	2000	15000	15000
3	3000	15000	15000
4	3000	21000	21000
5	4000	24000	24000

Tabla 2

De las simulaciones se graficaron: el tiempo de respuesta y las pérdidas de vencimientos de las tareas del sistema en forma de diagrama de barra. El tiempo de respuesta, en forma análoga a las simulaciones anteriores, se tomó como el promedio de los tiempos de respuestas de los DEV dentro de los 100000 µseg. de tiempo de simulación. En este caso se realizaron las simulaciones utilizado las tres disciplinas de prioridades enunciadas: EDF, SRTP y RMS.

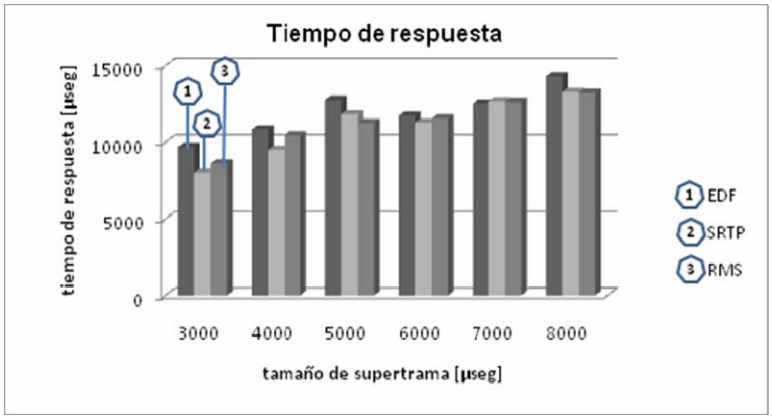


Figura 5: tiempo de respuesta

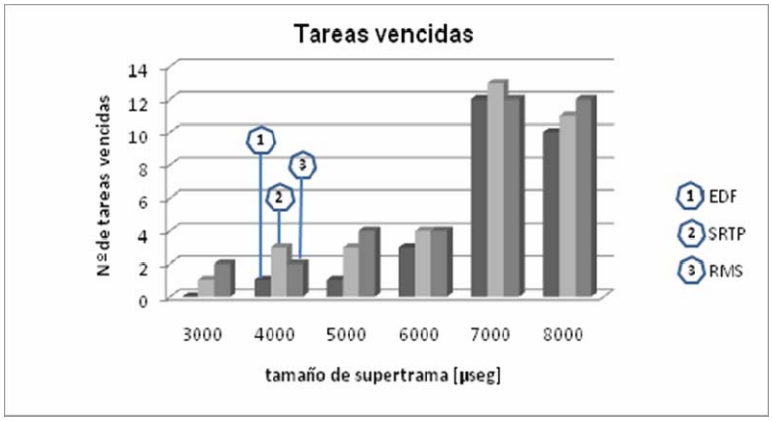


Figura 6: tareas vencidas

La figura 5 refleja el tiempo de respuesta comparativo entre las distintas disciplinas de prioridades, donde se aprecian las diferencias en los resultados obtenidos. Se puede observar que con la disciplina SRTP el tiempo de respuesta del sistema es, en general, menor que los otros dos esquemas de prioridades. Esto está ligado a la relación entre el C, T y D, ya que como SRTP ordena por tiempo de ejecución, ésta logrará los mejores tiempos de respuesta. En cambio EDF tendrá la menor cantidad de tareas vencidas, ya que su ordenamiento es por menor tiempo de vencimiento.
La figura 6 muestra que para los valores máximos de tamaño de supertrama simulados (7000 µseg y 8000 µseg), se produce un fuerte incremento en tareas vencidas, producto de la conjunción de los valores de los parámetros seleccionados en la tabla 2 y tamaño de supertrama.
El último escenario seleccionado se muestra en la tabla 3. Este patrón de datos se parametrizó con tareas de distinta duración (C), pero de igual períodos de arribo (T) y vencimientos (D)

Tarea	C [µseg]	T [µseg]	D [µseg]
1	3000	33000	33000
2	2000	33000	33000
3	5000	33000	33000
4	3500	33000	33000
5	5500	33000	33000
6	6500	33000	33000
7	1000	33000	33000
8	1000	33000	33000
9	1000	33000	33000
10	1000	33000	33000

Tabla 3

Las simulaciones fueron realizadas para distintos tamaños de supertrama, las mismas fueron tomadas desde 400 µseg hasta 2000 µseg en pasos de a 400 µseg., tal como se muestra en las figuras 7 y 8. El tiempo de cada simulación fue de 100000 µseg. para cada tamaño se supertrama, manteniendo en 10 a la cantidad de nodos de la “piconet”.
De las simulaciones graficamos el tiempo de respuesta y la pérdidas de vencimientos de las tareas del sistema. El tiempo de respuesta fue tomado como el promedio de los tiempos de finalización de las tareas de los nodos que componen la piconet dentro de los 100000 µseg. Las simulaciones se realizaron sobre las disciplinas EDF y SRTP. De la figura 7, se puede observar como mejora el tiempo de respuesta si la disciplina de prioridades utilizada es SRTP. Asimismo, la figura 8, muestra a la disciplina SRTP como una mejor opción a la hora de seleccionar un esquema de prioridades. Esto se debe a que la disciplina SRTP ejecuta primero las tareas de menor C, mejorando el tiempo de respuesta y dado que existen muchas tareas con bajo valor de C, esto provoca una menor cantidad de tareas vencidas. EDF ejecuta primero las de menor vencimiento y en este caso son las de mayor C con lo cual origina un mayor tiempo de respuesta y la perdida de vencimientos.
De la figura 8 se observa que EDF tiende a mejorar a medida que el tamaño de supertrama aumenta, tendiendo a igualar los resultados del esquema SRTP. Esto se debe a que a mayor tamaño de supertrama, la incidencia de los tiempos destinados al “beacon” y los MCTAs es menor.

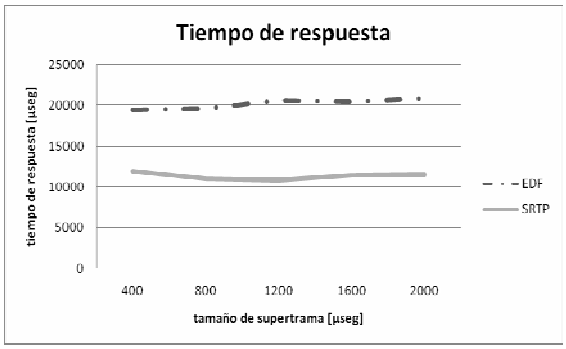


Figura 7: tiempo de respuesta

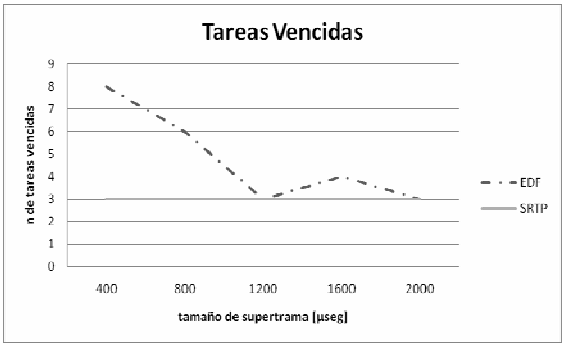


Figura 8: tareas vencidas

5 Conclusiones

El Estándar 802.15.3 especifica la capa física (PHY) y el control de acceso a medio (MAC) en redes inalámbricas de área personal, que permite transmitir datos a alta tasa de transferencia y mínimo consumo.

En este trabajo se analizó, mediante simulaciones, el comportamiento de la capa MAC del estándar 802.15.3 bajo distintas disciplinas de prioridades como EDF, SRTP y RMS, con el fin de evaluar la mejor alternativa de tamaño de supertrama, utilizando como métrica el tiempo de respuesta y el número de tareas vencidas (datos perdidos).

El análisis se realizó variando el tamaño de supertrama, pero se mantuvieron fijos para cada una de las corridas de las simulaciones. Además se utilizó los MCTAs para solicitar el tiempo de uso del canal, en lugar del CAP, como alternativa de la norma.

Para la evaluación, se ensayaron tres tipos de escenarios, donde se variaron tamaños de supertrama, cantidad de nodos y tipo de tráfico para las disciplinas de prioridades EDF, SRTP y RMS.

Del primer escenario, al tener las tareas el mismo tiempo de duración y el mismo tiempo de generación, no hubo diferencias entre las distintas disciplinas de prioridades. Si se pudo notar que la elección de un valor de supertrama menor da un mejor rendimiento tanto en el tiempo de respuesta con el número de tareas vencidas.

Del segundo, al tener tareas de distintos valores de duración y tiempo de generación, podemos comparar los tres esquemas de disciplinas obteniendo que, eligiendo un valor adecuado de la supertrama podemos mejorar sustancialmente el rendimiento de la red.

Por último se tomó tareas de distinto tiempo de duración e igual tiempo de generación donde vimos que SRTP obtuvo mejor tiempo de respuesta para cada uno de los tamaños de supertrama y el menor número de tareas vencidas.

Podemos concluir que una de las consideraciones más importante a tener en cuenta es el tamaño de supertrama que se seleccione, el cual estará vinculado al tipo de tráfico de cada caso en particular.

Mantener una estructura de supertrama fija en redes con tráfico variable, implica que se desperdicia el ancho de banda disponible, pues puede suceder que quede tiempo ocioso en la supertrama.

En los futuros trabajos se planteará trabajar con tamaños de supertrama que varíen dinámicamente durante el proceso de transmisión dependiendo de diversos parámetros de la red. Por otra parte se trabajará en la elaboración de un algoritmo que permite la recolección de los parámetros de los DEVs de tal forma que no se malgaste la utilización del tiempo del canal.

6 Referencias

[1] Sun-Myeng Kim, Young-Jong Cho, “Scheduling Scheme for providing QoS to Real-Time Multimedia Traffics in High-Rate Wireless PANs”, IEEE Transactions on Consumer Electronics, Vol. 51, No. 1160 . 4, NOVEMBER 2005.

[2] Attila Torok, Lorant Vajda, Attila Vidacs, Rolland Vida, “Techniques to improve Scheduling Perfomance in IEEE 802.15.3 based ad hoc networks”, IEEE GLOBECOM 2005 proceedings.

[3] Yi-Hsien Tseng, Eric Hsiao-kuang Wu, Gen-Huey Chen, “Maximun Traffic Scheduling and Capacity Analysis for IEEE 802.15.3 High Data Rate Mac Protocol”, IEEE VTC 2003-Fall, vol. 3, pp. 1678-1682, Oct. 2003.

[4] IEEE, “Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for High Rate Wireless Personal Area Networks (WPANs)”, IEEE std 802.15.3 – 2003, Sept. 2003.

[5] A. Török, L. Vajda, Y. Kyu-Jung, J. Sun-Do, “Superframe formation algorithms in 802.15.3 networks”, Proc., IEEE Wireless Communications and Networking Conference (WCNC), Atlanta, Georgia, USA, 2004.

[6] L. Vajda, A. Török, L. Kyu-Jung, J. Sun-Do, “Hierarchical superframe formation in 802.15.3 networks”, Proc., IEEE International Conference on Communications (ICC), Paris, France, 2004.

2007 Argentine Congress on Computer Science (Congreso Argentino en Ciencias de la Computación - CACIC 2007)

Capa Física y de Enlace de la Tecnología PLC (Physical and Link Layer in Power Line Communications Technologies)

Jorge García Guibout and **Carlos García Garino**
Instituto Tecnológico Universitario, Universidad Nacional de Cuyo,
Mendoza, Argentina, jgarcia@itu.uncu.edu.ar

and

Rubén Jorge Fusario and **Antonio Ricardo Castro Lechtaler**
Universidad Tecnológica Nacional – Escuela Superior Técnica – IESE,
Buenos Aires, Argentina, rfusario@speedy.com.ar and acastro@utn.edu.ar

and
Guillermo Sevilla
CITEFA - Escuela Superior Técnica – IESE,
Buenos Aires, Argentina, gsevilla@citefa.gov.ar

ABSTRACT

PLC technology[1] - Power Line Communications- refers to the transmission of data using the domestic as well as the low-voltage electrical net. Depending on the type of network used as support it is divided in PLC indoor and outdoor. The first refers to the use of the domestic network and uses higher frequencies (5 MHz to 30 or 40 MHz). Outdoor PLC uses the distribution network and low frequencies of 1 MHz to 5 or 10 MHz. It employs the OFDM multiplexation technique- Orthogonal Frequency Division Multiplexing- which divides the frequency spectrum in narrow channels with independent carriers. At the level of media access protocol, it generally uses those which are compatible with the IEEE 802 norm, with controlled access to avoid collisions. OFDM modulation achieves transmission speeds which are suitable for multimedia services and applications as well as those of critical mission currently used, and with security conditions when facing a noisy channel such as the power network.

KEYWORDS: PowerLine Communications, OFDM, MAC, CSMA/CA, Last Mile.

1. INTRODUCTION.

Currently, the market offers a wide range of last mile technologies. Not only will technology provide telephone services, but also broadband services, offering added value to providers and users.

[1] Also known as PLT - PowerLine Telecommunications- o BPL -Broadband PowerLine.

A variety of connectivity models can be developed based on the different technologies aiming to satisfy user's needs, interconnecting voice and data networks, or both services separately. We could mention, for instance, Voice over IP, TV or Video on Demand, analogical or digital, low cost local wireless links. This work deals with PLC technology – PowerLine Communications- which uses the low voltage domestic electrical distribution network as a support for data, Internet, voice, video, and video-on-demand transmission.

Experimental systems which use power grid for high speed data transmission have been developed at the telecommunications level. The development possibilities for this technology are significant considering the rising demand of broadband internet services.

A straightforward advantage is the possibility to offer telephone service, internet access, interactive TV, domotics, and others wherever electrical power is present. More than 3.5 billion people have electrical power at home, as opposed to 1.8 billion who have access to telephone services. Practically every room of a house or store has an electrical outlet for 220 VAC or 110 VAC.

Additional advantages are the rapid expansion (no need to build ducts if compared to cable operators), and the fact that is does no require a frequency assignment, as opposed to wireless technologies.

The most important advantages of PowerLine Communications are worldwide availability, low cost, and easy installation. In addition, it cannot be ignored the convenience of connecting any device to an electrical outlet which would allow to surf the web, receive videos, transmit data, and speak over the phone.

Since 1997, United Utilities from Canada and Northern Telecom from England have introduced a technology which could give Internet access through power lines: PLC – PowerLine Communications. By late 1999 and the beginning of the year 2000, Spain also joined this trend with Endesa. The U.S. has also developed several initiatives; although they never accomplished satisfactory commercial results. Nevertheless, ever since then, electrical companies have been interested in taking better advantage from their networks through numerous projects to develop communication services.

On the other hand, big technological corporations such as 3COM, AMD, Cisco Systems, Compaq, Conexant, Enikia, Intel, Intellon, Motorola, Panasonic, Diamond Multimedia, and RadioShack created and alliance called ***HomePlug PowerLine***. The main objective was to create devices and to promote the rapid growth of the technology in homes.

2. PLC TRANSMISSION

As mentioned, PLC is implemented in two forms: ***indoor systems*** also known as ***last meter or last inch***, which use the users´ indoor network, and the ***outdoor systems*** which work with the electric distribution grid in the ***last mile.***

To operate simultaneously on both systems, different carrier frequencies are used over the same power lines. Usually, last mile systems use low frequencies to the loss of signal due to attenuation. Meanwhile, in the case of a building interior, higher frequencies can be used because the higher attenuation from high frequency use is compensated by the lower distances the signal travels.

For a long time, power companies have used networks, transmission and distribution, to send electricity and telecommunication signals. Consequently, PLC has been used for a while, but only for the transmission of control signals with very low bandwidth. These networks are used for monitoring and long distance control as well as supervision from equipment used in electrical distribution[2]. With these systems, power companies have an extremely cheap means of sending monitoring and control signals.

The method used is based on the transmission of multifrequency signals in the electric grid. Frequencies are selected in such a way that it is possible to separate them clearly to avoid interference. Frequencies of 10 to 450 kHz have been commonly used to communicate services such as ***carrier wave***, while electric signals work in a range of 50 to 60 Hz.

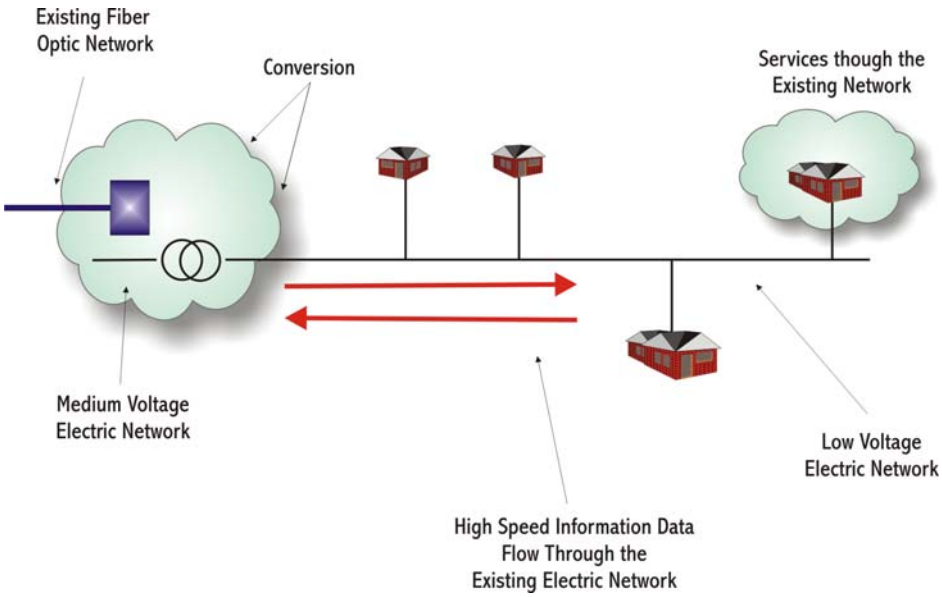


Figure 1: Last Mile PLC Diagram

Two fundamental elements lay in the originality of current PLC technology compared to the one which has been in use previously:

- The use higher frequencies.
- The ability to accomplish broadband transmission.

The frequencies vary from 1,7 to 30 MHz. With higher frequency use, higher transmission speeds can be reached.

However, the main disadvantage of PLC transmission lies in this use of high frequency signals for data transmission because they will not go through the ***Distribution Transformers*** - ***MT/BT***. They become like a low frequency filter with maximum cut at 20 KHz.

Thus, data signals must be separated from the electric signals with voltage transformers. At this point, signals also enter a microwave or optic fiber backbone. It is a valid technology transmitting high frequency signals for ***indoor*** or ***outdoor*** use.

[2] Such as measuring devices, switches, failure detecting devices, and so on.

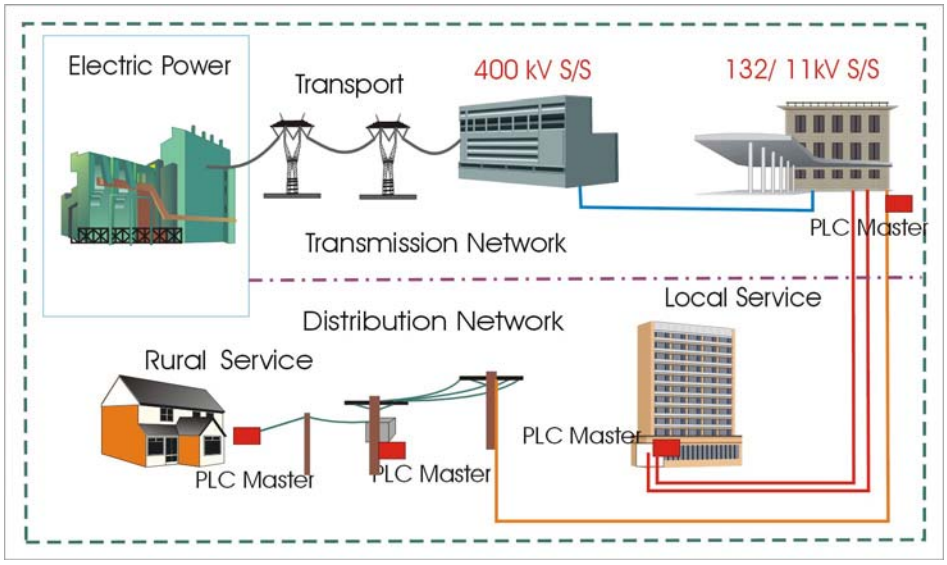


Figure 2: Diagram of a PLC Medium and Low Voltage Network

Figure 2 shows the diagram of a PLC network in which medium and low voltage is contemplated. Figure 3, on the other hand, depicts a diagram of PLC distribution in a user's building.

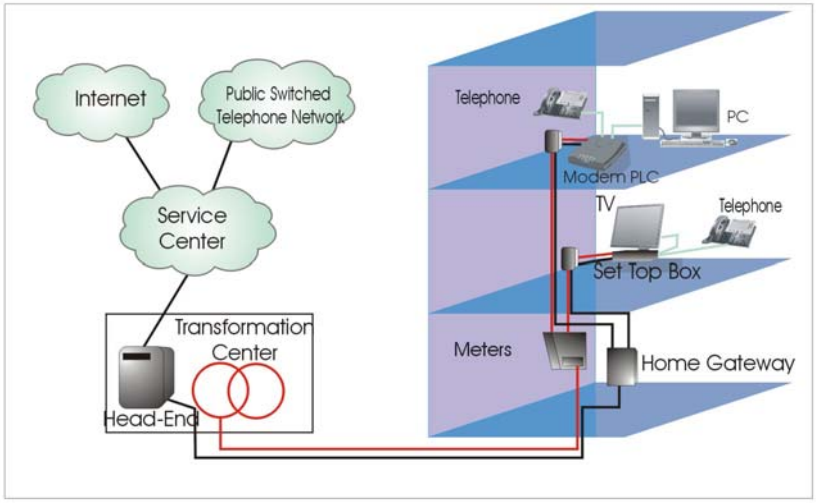


Figure 3: Diagram of PLC Distribution in a User's Building

In brief, the following considerations describe the architecture of this technology. Energy reaches the user as low frequency alternate current, 50 to 60 Hz, while data traveling through the same power grid using the bandwidth of 1 to 34 MHz, through the following steps:

- High Voltage, from 220 to 400 Kilovolts, which carries energy to the transport substation
- Medium Voltage, from 66 -132 Kilovolts, up to the distribution substation.
- Medium Voltage, from 10-50 Kilovolts, up to the distribution center.
- Low Voltage, from 220-380 volts, which distributes the energy in buildings and homes

3. RELEVANT PLC TECHNICAL INFORMATION

If we analyze the structure of communications from the OSI viewpoint, we can se that the physical and link layer show differences. Higher layers do not vary from the traditional models. Regarding the physical layer, considerations need to be made on the design for PLC modulation. Finally, the link layer develops technologies for the MAC sub layer available.

3.1 Physical Layer: Modulation Diagram

Distribution grids are subject to a wide range of noise. If it is greater than the signal level, the latter cannot be recovered properly and might be lost. Henceforth, simple modulation designs are not appropriate for transmission over power grids (for example FSK or PSK). The two most useful methods are: ***OFDM - Orthogonal Frequency Division Multiplexing*** and ***DSSS - Direct-Sequence Spread Spectrum***.

The main advantages of DSSS are the lack of sufficient bandwidth and the fact that the power network does not have a plain transmission characteristic in the entire assigned bandwidth. Consequently, higher levels of attenuation are found in some frequencies rather than others.

The other alternative to use in the physical layer is OFDM. An optimum design could result from OFDM at current conditions or in the future. OFDM is a well proven technique, used in applications such as ***DAB - Digital Audio Broadcasting***, and in ***ADSL - Asymmetric Digital Subscriber Line.*** OFDM will most likely be used in Digital Television standards, as would be ***DVB - Digital Video Broadcasting***[3]. OFDM is related to the ***FH technique – Frequency Hopping*** of expanded spectrum, showing strengths when considering interferences and multiple accesses.

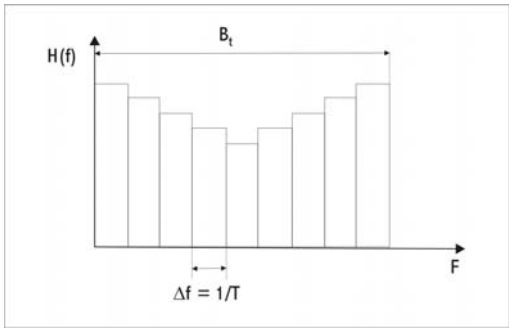


Figure 4: Bandwidth Subdivision

In OFDM, the available frequency spectrum (B_t) is segmented in several narrow channels. A data stream is transmitted by frequency multiplexation using N carriers with parallel frequencies f_1, f_2,... f_N. Figure 4 depicts the sub channels where each one will have a bandwidth equal to:

$$\Delta f = \frac{B_t}{N}$$

[3] Known as European standard

Attenuation and group delay are constant en each channel. Figure 5 shows the OFDM diagram and advantages.

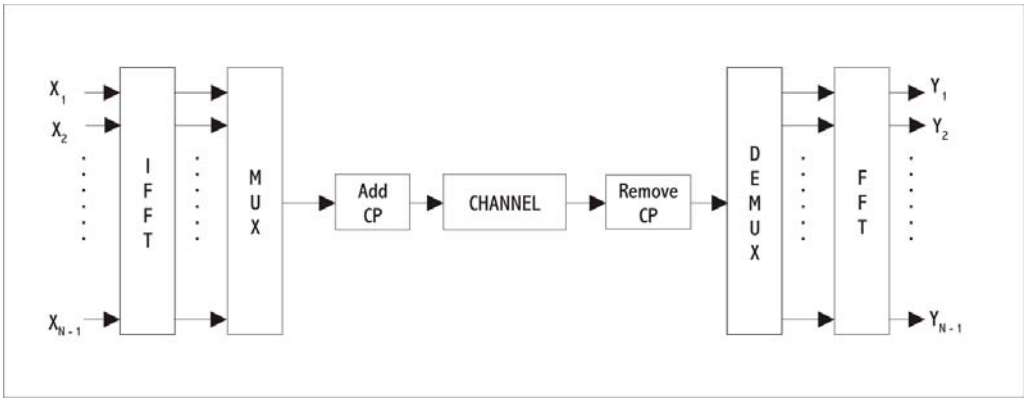


Figure 5: OFDM Diagram

Advantages:

- Dispersion effect decrease in time domain.
- Interference reduction among narrow bands
- High efficiency in the use of the bandwidth.
- Transmission speed scalability for growing transmission rates.
- Flexibility and adaptability according to modulation, data rates and bit charge.
- Excellent ICI response
- No need for channel equalization
- Not requiring hook-up in the oscillators phase

In this way, equalization, understood as the signal distortion compensation, has the advantage in OFDM that its circuits are simple. Besides, OFDM avoids extreme conditions, such as high frequency attenuation, narrow band interferences, and selective vanishing due to multipaths. This is the result from taking each carrier as a narrow band signal with a modulation speed instead of a broadband signal with a high modulation speed.

Low modulation speed allows for a carrier interval which eliminates interference between symbols or carriers (ICI). Equalization, in this case, will be the inverse of the channel transference function. With a known sequence, it is possible to determine the channel transference function. Afterwards, the inverse of that function is calculated to obtain the equalization values.

A typical OFDM signal in the domain of time, S_{OFDM} (t), can be depicted as follows:

$$S_{OFDM}(t) = A.\ \text{rect}\left[\frac{t}{T}\right] \cdot \sum_{i=1}^{N} sen\left[2\pi\left(f_0 + \left[i - \frac{N+1}{2}\right] \cdot \Delta f\right) \cdot t\right]$$

The minimum frequency spacing is equal to $\Delta f = 1 / T$, where T is the period of the wave. This equation describes a frequency pack in the range:

$$f_0 - [(N-1)/2].\ \Delta f = f_0 - (B_t - \Delta f) / 2 \text{ up to } f_0 + [(N-1)/2].\ \Delta f = f_0 + (B_t - \Delta f) / 2$$

Figure 6, illustrates the spectral condition for N= 7

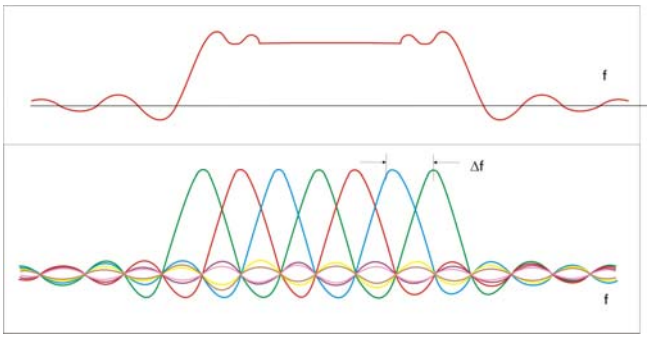


Figure 6: Spectral Distribution of 7 channels in OFDM [1]

An excellent use of the spectrum is achieved in this way, enabling the use of high speeds. It has a double characteristic: low passing and at the same time a filtering effect limiting the range of usable frequency. Frequencies in OFDM are transmitted sequentially. The main difference is that each carrier is modulated, carrying a piece of the data package, and that a great number of carriers, hundreds generally, are transmitted in a parallel way.

The transmitted signal will be the sum of many modulated carriers. The synthesis of an OFDM signal can be carried out by the Inverse of Fourier Discrete Transfer, and in the receptor, the complementary operation from this function will be carried out.

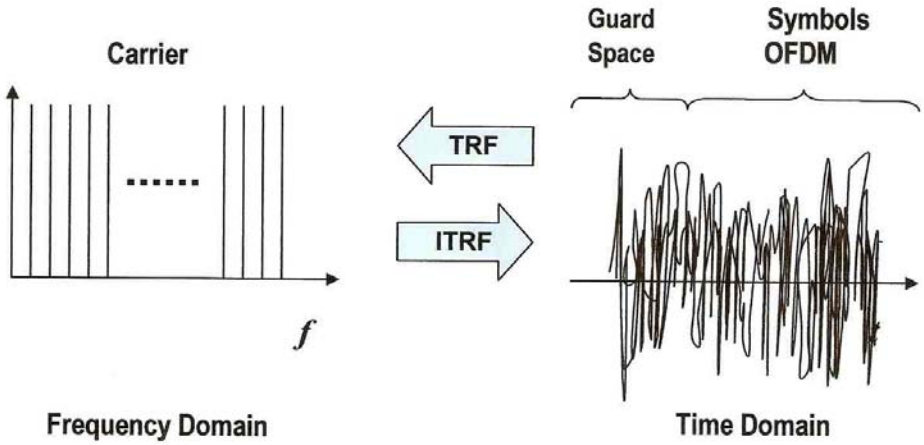


Figure 7: Same Signal Representation in Time and Frequency Domain through Fourier Transform

The complex processes of modulating and demodulating thousands of carriers simultaneously are fortunately equivalent to the operations in a Fourier Discrete Transform, FFT. An algorithm has been developed, called Fast Fourier Transform, FFT. In this way, the implementation of OFDM demodulators allows for the existence of series receptors.

As mentioned before, each OFDM carrier transports part of the data stream. Data does not need to be equally distributed among carriers; nor does it need the same modulation process. Considerable advantages can be obtained from the properties of the channel. If the transmission band has low attenuation and low interference levels, the carriers in that range can be modulated with complex modulation processes such as QAM. In other parts, where the values of the signal/noise relation are deficient, BPSK or similar might be used.

Note that IFFT, as well as FFT, take significant processing and must be carried out in less than T time. Therefore, modern digital signal processors need to be used. This also improves transmission speed. Finally, HomePlug has chosen OFDM as the core technology for PLC high speed networks which could lead them to great success [5].

A brief summary of OFDM technology follows:

- The idea of dividing the spectrum in several narrow bands.
- Frequencies are superposed and orthogonal to obtain spectrum efficiency.
- For each sub-band, well known modulation techniques are used, such as BPSK, QPSK y QAM.
- Equalization is reduced by placing sub-carriers very close to each other.
- OFMD modulation is generated by FFT process – Fast Fourier Transform.
- Equalization is reduced, placing sub-carriers close together:

  $$M = N \times B$$

  Where:
  - ✓ M = quantity of coded bits
  - ✓ N = quantity of sub. carriers
  - ✓ B = number of bits per modulation symbol.

  Ex.: B=2 in QPSK

- The FFT inverse is calculated to obtain several sub-carriers which altogether will determine a domain in simple time.
- A cyclic prefix is added..
  - ✓ It eliminates carrier interference maintaining orthogonality.
  - ✓ It removes Inter Symbolic Interference.
- On the other end, CP is extracted and FFT is applied to later decode, analyzing the different phases and amplitudes of sub-carriers.
- To eliminate multiple path reflexes, interleaving and error correction techniques may be used.

3.2 MAC Technology in PLC:

Medium Access Layer – MAC, is used to transfer data between the physical layer and the LLC sublayer of the link layer. Given that the power grid is ***shared***, a special organization for access is required. Manufacturers and cooperating organizations in PLC standardization processes do not use standards for the MAC layer yet.

For ***outdoor*** systems, HomePlug PorweLine Alliance has not set standards yet; although it is expected to be done this year. In the ***indoor*** case, a MAC layer standard has been established: the use of ***CSMA/CA - Carrier Sense Multiple Access / Collision Avoidance***.

The level of noise in PLC is significantly greater then in Ethernet. Thus, ***collision avoidance*** instead of ***collision detection*** becomes relevant - the latter assumes that each node can listen to every other node.

Another alternative is **CDMA – Code Division Multiple Access**, considered more appropriate for PLC due to its robustness when facing noise and unwanted signals. It allows a station to transmit at all times and through the entire frequency spectrum. It separates multiple and simultaneous transmissions using codification theory. The technique does not need global coordination or synchronicity. Supposedly, the streams that collide do not suffer distortions but multiple signals are added in a linear fashion. CDMA technology is based on data transmission followed by a bit stream or code recognized by the receptor. Henceforth, the receptor carries out a Fourier transform to detect data. In this way, several channels can transmit at the same time and the same frequency but each carrying a different code.

Some manufacturers have developed their own techniques to access the medium such as DS2 [3] and Plus Net de Maint.net [8]. They use intelligent repetition which, aided by distribution network attenuation, forms cells where different PLC systems work. Units use point-multipoint protocols of own design, resulting in a completely transparent IP system for the final user.

Note that presently there is no MAC protocol for the link layer, in this sense ETSI 102 249 V1.1.1 (2005-06) standard, paragraph 6, which points out that the MAC sub layer must follow the practices of 802.Consequently frame delimiter, origin, and destination identification, transmission access control have to be full filled. These conditions are basic requirements to access the medium but do not define a specific protocol.

A similar case can be argued for the OPERA Consortium ***Open PLC European Research Alliance*** [9], which presents a point multipoint protocol based on TDMA with a net administrator. For that purpose, ***TDD - Time Division Duplex***, or ***FDD - Frequency Division Duplex*** is used as established in the standard OPERA D47 [D]. The first shares the channel mainly in transfer systems of asymmetric information. The latter shares the channel with standard frequencies, especially appropriate for symmetric transfers. These techniques could handle the different types of service qualities considered by OPERA

4. STANDARIZATION

Standards such as HomePlug HP 1.0 apply OFDM modulation with a total of 84 carriers in the frequency range of 4,5 MHz a 21 MHz, and due to the need for channel separation in this frequency band 76 carriers are available, using differential PSK modulation, as shown in Table 1. ***ROBO - Robust-OFDM*** is a modulation design developed by HomePlug with information bit repetition in each OFDM symbol, especially for specific information like broadcast packages which cannot be transmitted in the chosen modulation for each network member.

The standard established a protocol ***CSMA/CA*** link layer, determining two ways of medium occupation: ***PCS – Medium Carrier Detection, on the physical layer,*** and ***VCS – Virtual Carrier Census*** which is established in the thread head and processed at the link layer level. This enables the emission of data burst from a network component to complete data transmission.

Modulación		Corrección de error hacia adelante	Velocidad de la Capa Física en Mbps
DQPSK 3/4	DQPSK	Código de convolución 3/4	13,78
DQPSK ½	DQPSK	Código de convolución ½	9,19
DBPSK ½	DBPSK	Código de convolución ½	4,5
ROBO	DBPSK	Código de convolución ½ y cada bit es repetido cuatro veces	1,0

Table 1: Modulation Diagram And Error Correction

A priority mechanism is also set through the Standard, following the 802.1Q norm, VLAN tags, where the tagged threads inform the priority level.

This requires a priority resolution period which takes place after obtaining the thread reply with the data received free of errors. Retransmission is not necessary. At that point, there are two competence periods. In the first, higher priority threads compete; lower priority threads will follow, allowing higher level traffic to interrupt lower level traffic burst. In this way, high quality of service (QoS) can be established for the traffic which needs low delay as voice or video[4].

5. ADVANTAGES AND DISADVANTAGES OF PLC

VENTAJAS	DESVENTAJAS
Low cost	Yield is subject to home appliance use
Use of existing cables.	Old cables might affect yield
Electric outlets are present in every room.	Data security is difficult.
Very easy to install	QoS issues when constant bandwidth is required.
Every data device requires connection to the electrical network, even without using this technology.	Scarce competition and limited equipment manufacturing

6. MAIN APPLICACIONS OF PLC

- **Telephones**
  In VOIP – Voice over IP – mode. Some modems incorporate the VOIP gateway. Systems are designed to guarantee quality of service.

- **Internet Access**
  Broadband internet access is ***the*** PLC application most requested from client viewpoint. The nature of traffic bursts enables an effective multiplexation in the available bandwidth.

- **Interactive Services (games, e-learning, and others)**
  Interactive services can comprehend all sorts of Media on Demand, such as Media, Video, Music, News, and so on, with different degrees of interactivity. In any event, bandwidth guarantee and adequate delay times are required. In the case of network games, no special demands might impose the need for a service design with players connecting to a ***Game Room***

[4] Isochronous traffic

and exchanging moving commands and actions over pre-established scenarios, except for maintaining low delay.

- **Others (Videoconferences, VPN, Peer to peer applications, and so on)**
  These services are commonly born at the IP level for authentification functions and security access which reside in the Broadband Access Servers.

7. CONCLUSION.

The most convenient modulation technique for this technology is OFDM with a carrier modulation of higher bit quantity to achieve greater speed, but accounting for the interference and noise level of the network.

Even when several corporations have developed their own link layer protocols, such as Intellon or DS2, all of them converge in following the 802.3 standard in search for compatibility and transmission efficiency among the different networks in a MAN or WAN.

Although this is valid for ***PLC indoor*** and ***PLC outdoor***, it should be noted that the critical technology is outdoor due to the structure of the distribution grid with a high number of clients and wide signal radios and because of the greater interference and higher potency levels.

These issues require modulation systems robust to noise levels, and consequently slower. When this is considered with the electromagnetic compatibility issues and the lack of standards and regulation, it is clear that the market tends to choose alternative technologies such as XDSL. Currently in our country, broadband Internet access is provided by ADSL Technologies, cable modem – CATV and optic fiber – FTTx, and for the moment in a lesser degree, through wireless technolgies[5]. In this context, PLC is a developing experimental stage technology.

The future of indoor PLC might not be the same because its advantages and the possibility to reach greater speeds allow it to compete with wireless links and wiring, especially in the SOHO market sector.

In sum, based on the use of the low voltage power grid, PLC technology offers a solution to geographic dispersion, nationwide service, integration of low income households, price according to income, broadband internet access and so on. This is why we wonder if PLC will provide the answers in the issues that other technologies failed; and if it will finally break social, geographic and technological barriers which have stagnated the development and expansion of telecommunication networks in unprotected and discriminated areas by lack of sufficient resources. May this technology grant more households in the country with broadband access services.

8. ACKNOWLEDGEMENTS.

The financial support provided by Agencia Nacional para la Promoción Científica y Tecnológica and CITEFA (Project PICTO 11-18621, Préstamo BID 1728 OC-AR) is gratefully acknowledged.

[5] Links via satellite, local WLAN networks - Wireless Local Area Network

REFERENCES.

[1] Doster, Klaus, T. PowerLine Communication. Prentice Hall PTR. 2000.
[2] The PowerLine Communications Association. www.plca.net.
[3] DS2, Fabricante de Circuitos Integrados. www.ds2.es.
[4] Stott, J. H. Explaining some of the magic of COFDM, Inform, Proceeding of 20th International Television Symposium 1997.
[5] Home Plug PowerLine Alliance (HPA). www.net-athome.com.
[6] European Telecommunications Standards Institute, ETSI. France. www.etsi.org.
[7] Iwan, Eduardo. AP Tecnologías de Convergencia sobre Redes de Video Cable y Redes Eléctricas. Expo Comm Argentina. Buenos Aires. 2006.
[8] Maint.net, www.main_net.com
[9] The Opera Consortium/Open PLC European Research Alliance. www.operaconsotium.com
[10] OPERA D47, "General Specification of the MAC Layer", OPERA IST Integrated Project No 507667 (funded by EC), deliverable report D47, WP5, V1.1, 26/07/2004.
[11] CISPR 22; "Limits and methods of measurement of radio disturbance characteristics of information technology equipment". CISPR Publications 22. Geneva, Switzerland. 1993.
[12] EN 55022: Equipos de tecnología de la información. Características de las perturbaciones radioeléctricas. Límites y métodos de medida coincidente con CISPR 22 [10]. CENELEC. Bruselas. 1998.
[13] ETSI TR 102 175 V1.1.1 (2003-03). PowerLine Telecommunications - PLT: Channel characterization and measurement methods.
[14] ETSI TR 102 324 V1.1.1 (2004-05). PowerLine Telecommunications - PLT; Radiated Emissions. Characteristics and Measurement Method of State of the Art PowerLine Communication Networks.
[15] EN 55011: Límites y métodos de medida de características de perturbaciones de radio de equipos de radiofrecuencia de aplicación industrial, científica y medica (ISM).CENELEC. Bruselas. 1999.

2007 Argentinean Congress on Computer Science (Congreso Argentino en Ciencias de la Computación - CACIC 2007)

Comportamiento de la Tecnología PLC en la Red Eléctrica (Power Line Communications in the Electric Network)

Jorge García Guibout and **Carlos García Garino**
Instituto Tecnológico Universitario, Universidad Nacional de Cuyo,
Mendoza, Argentina, jgarcia@itu.uncu.edu.ar

and

Rubén Jorge Fusario and **Antonio Ricardo Castro Lechtaler**
Universidad Tecnológica Nacional – Escuela Superior Técnica – IESE,
Buenos Aires, Argentina, rfusario@speedy.com.ar and acastro@utn.edu.ar

and

Guillermo Sevilla
CITEFA - Escuela Superior Técnica – IESE,
Buenos Aires, Argentina, gsevilla@citefa.gov.ar

RESUMEN

La tecnología PLC[1] - PowerLine Communications- está referida a la transmisión de datos utilizando la red eléctrica, tanto domiciliaria, como la red de distribución de baja tensión. Dependiendo del tipo de red que se utilice como soporte esta tecnología se divide en PLC indoor y outdoor. La primera se refiere a la utilización de la red domiciliaria y utiliza las frecuencias más altas de 5 MHz a 30 ó 40 MHz. La segunda, outdoor, usa la red de distribución y las frecuencias bajas de 1 MHz a 5 ó 10 MHz. El mayor inconveniente que presenta es su incompatibilidad electromagnética con los servicios inalámbricos ya existentes, como las emisoras de AM, BLU, radios aficionados y otras. En la modalidad outdoor, la Red PLC actúa como antena emitiendo una señal que interfiere los servicios mencionados. Estos, a su vez, actúan sobre ella como generadores de ruido. Las señales PLC deben, además, discriminar sus propias señales con el ruido generado por los equipos conectados a la red (motores, electrodomésticos, etc.).

PALABRAS CLAVE: PowerLine Communications, Redes de Computadoras, Última Milla, Compatibilidad Electromagnética.

1. INTRODUCCIÓN.

Actualmente el mercado de las comunicaciones ofrece una gran variedad de tecnologías para cubrir el último tramo entre los proveedores de servicios y sus clientes en lo que se conoce técnicamente como última milla.

[1] Es también conocida como PLT - PowerLine Telecommunications- o BPL -Broadband PowerLine.

Para ello se pueden elaborar diferentes modelos de conectividad basados en las últimas tecnologías de banda ancha que se pueden aplicar, pero en todos los casos se busca satisfacer la totalidad de las necesidades de un cliente, interconectar redes de voz y datos, o ambos servicios separados, basadas en tecnologías emergentes de telecomunicaciones, como podría ser por ejemplo voz sobre IP, enlaces locales inalámbricos de bajo costo y alta velocidad, etc.

Se hace difícil la toma de una decisión de cual es la tecnología más conveniente a instalar, analizando no solo la factibilidad y costos implícitos en la puesta en servicio, sino deben considerar prioritariamente las realidades socioeconómicas de cada región en particular y del país en general.

En este trabajo se tiene en cuenta la tecnología PLC - PowerLine Communications, que es la utilización de la red de distribución eléctrica de baja tensión y domiciliaria como soporte para la transmisión datos, Internet, voz, video, video a demanda, etc.

La ventaja más obvia de esta tecnología, que permite ofrecer telefonía, acceso a Internet, televisión interactiva, domótica, etc., es la ubicuidad de la red eléctrica. Más de 3.500 millones de personas disponen de energía eléctrica en sus hogares, frente a los más de 1.800 millones que disponen de conexión telefónica. Prácticamente en cada habitación de una vivienda o de un local comercial existe un toma corriente. También es de interés su disponibilidad mundial, efectividad del costo y facilidad de instalación. A la vez, la conveniencia de conectar cualquier dispositivo a través de un toma corriente de corriente permite navegar, recibir videos, transmitir datos y hablar por teléfono.

En el presente trabajo se hace un estudio de la Capa Física. En particular, interferencias, capacidad del canal, problemas electromagnéticos, como así también el marco regulatorio y los estándares de la misma.

2. COMPORTAMIENTO DE PLC EN LA RED ELÉCTRICA.

2.1 Descripción de las posibles interferencias.

Las redes de distribución eléctrica presentan un ambiente típico con interferencias, en especial para propósitos de comunicaciones. Esto incluye tanto la red outdoor, como la red indoor, pese a que se debe hacer una distinción entre la transmisión en una y en otra.

Las redes de media tensión están alimentadas por líneas de alta tensión sobre transformadores. Éstas a su vez, a través de transformadores de baja tensión, forman sectores para alimentar barrios o industrias. Hasta frecuencias de 20 KHz, los transformadores son buenas barreras. Desacoplan las interferencias de las redes de alta tensión, de las de media, y éstas de las de baja.

Una gran parte de las interferencias es causada por maquinarias y dispositivos eléctricos en su operación rutinaria. Hay muchos picos de tensión diferentes debido a un amplio rango de eventos de cambio. En particular, las redes son estructuras abiertas a las señales electromagnéticas. Esto hace que haya numerosas señales originadas por las radiaciones de servicios de radio, principalmente en el rango de ondas medias y cortas.

Cuando analizamos el espectro de las amplitudes de interferencia en un toma corriente, tres diferentes clases pueden ser identificadas: ruido de fondo, interferencia o ruido de banda angosta, y ruido impulsivo. Los mismos son:

- ***Ruido de fondo.***
  Esta clase de ruido es de naturaleza estocástica y se puede describir por su densidad de potencia espectral. Valores alto de densidad espectral son característicos en redes comenzando en la frecuencia de alimentación (50 Hz) a frecuencias alrededor de 20 KHz. A partir de ella se nota un decrecimiento de la densidad a medida que crece la frecuencia. En 150 KHz es normal que sea 1/1000 del valor medido en 20 KHz. Así, los valores de ruido de fondo son de baja densidad de potencia espectral. Este ruido es denominado ***Ruido Blanco*** o ***Gaussiano***.

- ***Ruido de banda angosta.***
  Valores de amplitud de forma aguda indican interferencia de banda angosta. Ellas ocurren solamente en un rango de frecuencia limitado, pero con valores de densidad de potencia altos. Esta interferencia por debajo de los 150 KHz, puede ser originada por los cambios de fuentes de alimentación, lámparas fluorescentes, convertidores de frecuencia, aparatos de televisión o monitores. En frecuencias mayores la interferencia puede ser también por estaciones de radio.

- ***Ruido Impulsivo:***
  Es caracterizado por periodos de muy corta duración de tensión de entre 10 y 100 μseg. Pueden alcanzar una amplitud mayor a 2 KV. Estos eventos independientes son causados por encendido o apagado de aparatos eléctricos. Son aperiódicos. Los impulsos periódicos son causados normalmente por controladores de fase, y ocurren en el cruce por cero de la tensión de red.

  Dependiendo de la duración de la interferencia, uno o más bits pueden ser corrompidos. Tales errores se deberían prevenir por un adecuado cambio de codificación.

  Analizar y modelar este tipo de ruido está sujeto a investigación, por esto se define los siguientes parámetros para su caracterización:

  – Amplitud del impulso A_i
  – Ancho del impulso t_w
  – Intervalo de tiempo entre impulsos t_a

  Estas variables aleatorias y sus propiedades deben ser investigadas por medio de mediciones a través de la adquisición automática de datos, y su evaluación estadística. Ellas permiten obtener una síntesis del ruido impulsivo, con más pasos para emular el canal powerline. Una vez obtenidas las variables por diferentes formas de captura, se aplicaría para su evaluación la Transformada Rápida de Fourier.

A frecuencia muy bajas, de 50 ó 60 Hz, los niveles de interferencia son causados por armónicas de estas frecuencias, y por otras, independientes de las frecuencias de la tensión de alimentación. Esto se sumará al ruido impulsivo. Las armónicas independientes de la frecuencia de la red son de valores arbitrarios a la frecuencia de alimentación. Sus principales fuentes son los motores. Esta es la razón por lo que esta interferencia es encontrada en los niveles de bajo tensión. Otra fuente de este tipo son los eventos de encendido y apagado de equipos, que causan un espectro de amplitudes decrecientes inversamente proporcionales a la frecuencia. Estos eventos se pueden observar hasta frecuencias muy altas.

Observando la figura 1, la curva 3 refleja el ruido de fondo medido en una red domiciliaria en reposo. Este ruido no representa un factor de interferencia crítica para todo tipo de transmisión de información. Sus niveles son relativamente bajos, tanto que se debió recurrir a los límites de sensibilidad del instrumento de medida utilizado.

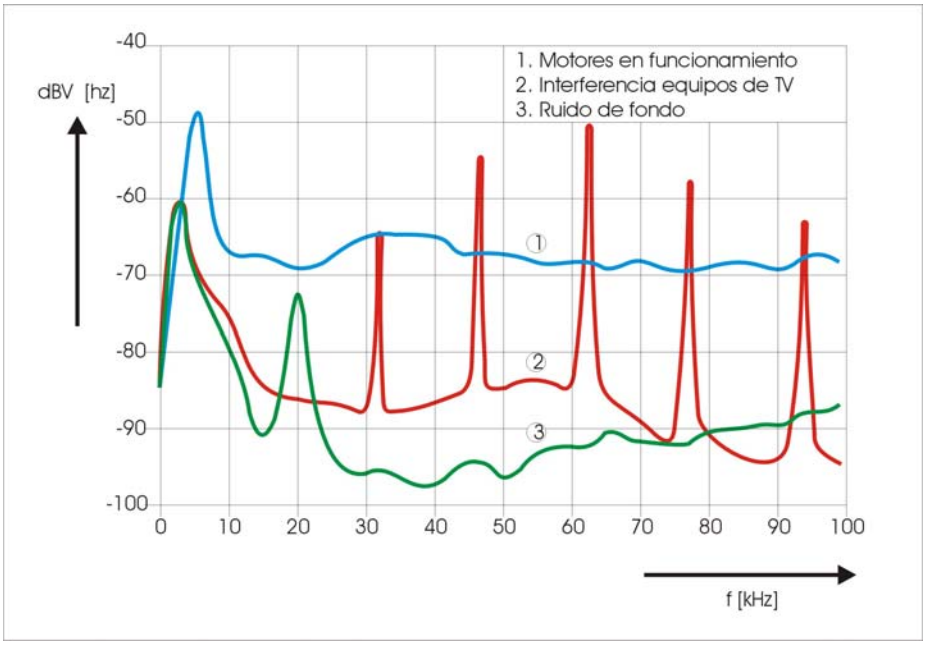


Figura 1: Posibles interferencias en una red eléctrica [1]

En contraste, las interferencias causadas por los motores, la curva 1 muestra que sus valores son más críticos. Estos son los usados en el hogar, en secadores de pelo, licuadoras, heladeras, etc. En el rango de frecuencias registradas es mucho mayor, que el ruido blanco en un valor de 20 dB mayor.

Cuando la fuente de esta interferencia está cerca del receptor puede bloquearlo. La curva 2 de la figura, se muestra la interferencia de banda angosta producida por un aparato de televisión. No se observa la frecuencia fundamental en 15,734 KHz, posiblemente por algún filtro del aparato. Sin embargo, las armónicas son de gran amplitud. Un sistema de transmisión que trabajara en ese rango de armónicas no hubiera podido funcionar mientras el televisor estuviera encendido. La ubicación de estas es constante, al punto que podría eliminarse ese rango de frecuencia.

La relación de la señal útil frente a la señal de ruido, como se sabe, se expresa en dB. Se la conoce como relación señal a ruido. En los enlaces de datos esta relación está directamente vinculada a la cantidad de bits erróneos frente a la cantidad de bits enviados. Es lo que se conoce como VER – Bit Error Rate.

2.2 Capacidad del canal de Powerline.

De acuerdo con la Ley de Shannon la capacidad de un canal con ruido, es decir la cantidad máxima de información en bps que es posible transferir resulta:

$$C = B \cdot \log_2 \left(1 + \frac{S}{N} \right)$$

C indica la máxima velocidad de transmisión de datos en bps, y *B* es el ancho de banda del canal. *S/N* es la relación señal a ruido del mismo.

Esta fórmula no es aplicable directamente a un canal de PowerLine, ya que la relación señal a ruido no es constante con en el ancho de banda *B*, ya que puede variar sustancialmente. Sin embargo, en la práctica la densidad de potencia de señal transmitida S_{rr} *(f)* y la densidad de potencia de ruido S_{nn} *(f)* son dependientes de la frecuencia.

Se los puede tomar dentro de un rango, por lo tanto la formula se puede modificar:

$$C = \int_{fu}^{fo} B \cdot \log_2\left(1 + \frac{Srr(f)}{Snn(f)}\right) df \quad \text{con } B = f_0 - f_u$$

Para obtener la expresión anterior debemos conocer el espectro de densidad de potencia de señal transmitida S_{rr} *(f)*. ***Esta será modificada por el comportamiento del canal***, el cual representamos por la función de transferencia *H (f) y la densidad de potencia de ruido* S_{nn} *(f).*

$$S_{rr}(f) = Srr(f) \cdot |H(f)|^2$$

La función transferencia, como la densidad de potencia de ruido son características fijas en un canal. Ellas son obtenidas por mediciones efectuadas sobre él. Por lo tanto, las únicas variables que quedan es la densidad de potencia transmitida, la cual está determinada principalmente por el esquema de modulación aplicado.

Luego, se podría llegar a la conclusión que altas velocidades de datos podrían ser alcanzadas aún en canales malos, si la densidad de potencia no fuera limitada. Sin embargo los requerimientos de compatibilidad electromagnética entre PLC y otros servicios inalámbricos establecen limitaciones en el rango de interés de alta velocidad de PLC, por ejemplo entre 1 MHz y 30 MHz.

Las mediciones en numerosos canales, a través de la evaluación de las formulas anteriores, indican que el ancho de banda disponible *B*, representa el recurso más importante para lograr altas velocidades. Como el ancho de banda en la práctica está fragmentado, diferentes canales de ancho variable serían distribuidos en el rango de 1 MHz a 30 MHz, obligando a elegir un esquema de modulación que permita un máximo aprovechamiento del mismo.

La capacidad del canal decrece con la distancia debido al comportamiento pasa bajo del canal powerline. Por ejemplo, para distancias de 100 m teóricamente excede los 250 Mbps, aun en distancias de 300 m y restringido a una rango de 5 MHz se puede obtener 14 Mbps. Pero se debe tener en cuenta que en enlaces de más de 200 m en la mayoría de los casos las frecuencias por arriba de 10 MHz no contribuyen en nada.

Esto apunta necesariamente a incluir esquemas de modulación complejos como ***QAM – Quadrature Amplitude Modulation***, obteniendo mejores resultados en esquemas de ***modulación adaptativa***. Ella optimiza el espectro disponible pues dividir a este en subcanales delgados, los cuales pueden transferir una parte del stream de datos, individualmente adaptada a su calidad.

2.3 Características de transmisión en alta frecuencia.

2.3.1 Característica del canal outdoor.

Mediciones efectuadas sobre red de distribución han permitido llegar a la conclusión que esta puede ser explotada para propósitos de telecomunicaciones hasta la frecuencia de 10 MHz aproximadamente. Canales de PowerLine disponibles a tal efecto se encuentran tanto en cables subterráneos, como cables aéreos.

Se ha observado un efecto de filtro en cables de áreas residenciales densamente pobladas. Allí, al haber gran cantidad de derivaciones a los hogares, y siendo estos de longitud constante, se produce una característica de FIR - Filtro de Respuesta Impulsiva. Ello hace que se supriman completamente rangos de frecuencias por arriba de unos pocos MHz.

Las investigaciones confirman que la velocidad de los datos puede estar en el orden de los Mbps, aún en enlaces críticos. Los distintos tipos de cables no exhiben un comportamiento completamente diferente, pero sus características más importantes se las pude describir con pocos parámetros. Otro resultado importante es la posibilidad de transmitir señales con niveles de potencia moderados. Esto mejora los problemas que se presentan, a causa de fenómenos de compatibilidad electromagnética.

En la Figura 2, se muestra el resultado de la medición de atenuación y ruido en un cable de tierra de 300 m en un rango de frecuencia de 500 KHz a 20 MHz. Se puede observar, que la atenuación crece con la frecuencia. También un comportamiento pasa bajo, condición que es común a todo cable de conexión a tierra. Puede verse esto, como una propiedad fundamental para frecuencias por arriba de 500 KHz.

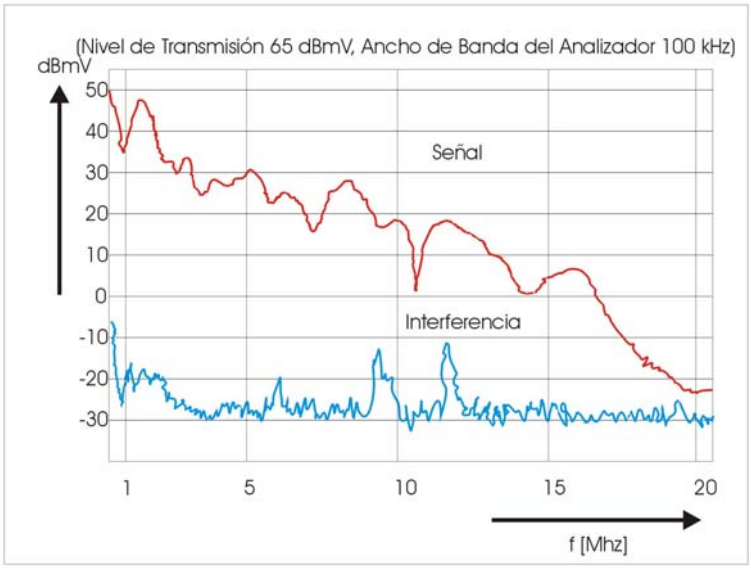


Figura 2: Atenuación y ruido en un cable de tierra de 300 m

En la misma figura, se pude observar en un rango extendido, la existencia de ruido de fondo, claramente por debajo de la señal recibida. La relación S/R se ha medido en el orden de los 50 dB, en el rango de 500 KHz a 6 MHz.

En las cercanías de los 17 MHz, se pude observar que la relación es de 20 dB, lo cual es suficiente para transmisiones de alta velocidad sin perturbaciones.

En el rango de 9 a 12 MHz, se observan picos de banda angosta. Ellos corresponden a las señales de difusión de onda corta. Esto lleva a que niveles de transmisión de 65 dBmV, equivalente a 1,77 V que garantizan una comunicación powerline, puedan perturbar la recepción de las señales de radio.

Las condiciones representadas en la Figura 3, representan un canal muy favorable. La calidad de los ellos varía enormemente, y uno de los parámetros que interviene es la longitud del cable. En esta figura se representa la atenuación de cables de distintas longitudes.

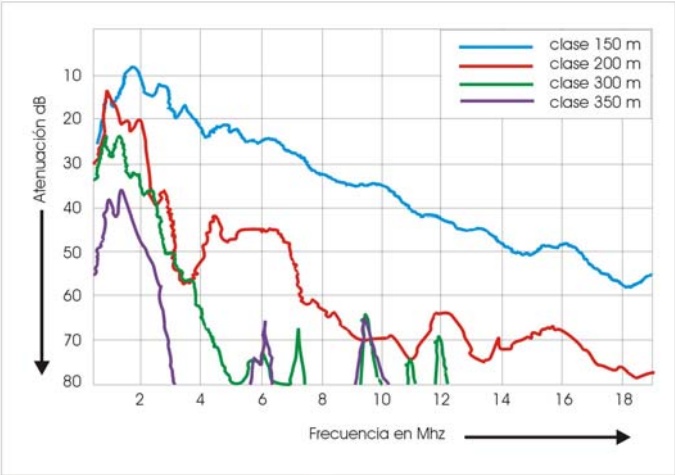


Figura 3: Atenuación y ruido en cables de distinta longitud.

La curva obtenida para la atenuación en cables cortos, del orden de los 150 m, presenta una característica pasa bajo y niveles de atenuación por debajo de los 60 dB hasta los 20 MHz.

Esto nos indica, que pueden establecerse comunicaciones sin problemas, aún con modelos de modulación simple y bajos niveles de transmisión. Para los cables de 200 m la atenuación crece significativamente, particularmente más de 25 dB. Esto sucede, quizás, debido a que en esta distancia haya muchos más clientes conectados, que en el cable de 150 m.

Dos casos muy extremos son los cables de 300 m y 380 m. allí se ve que el comportamiento pasa bajo está limitado a frecuencias muy bajas. Esto debido a un mayor número de conexiones domiciliarias que hace se comporte como un filtro de respuesta impulsiva finita.

Del análisis se desprende que, el ancho de banda disponible para PLC outdoor está por debajo de los 4 MHz. Además, las frecuencias bajas deben ser utilizadas en los enlaces de mayor distancia.

2.3.2 Comportamiento del canal indoor.

La Figura 4, muestra la curva de atenuación y ruido de una instalación domiciliaria de un edificio. Se observa, que el ruido también está presente en este tipo de instalación, y que tiene las mismas características que el canal outdoor. Además el mismo tiene un rango de atenuación de 40 a 80 dB.

Al comparar las curvas de atenuación del canal outdoor, con las indoor, a pesar de la corta distancia de este último, se puede ver claramente en las curvas de atenuación que se pierde la característica pasa bajo. Esto, obviamente es debido a su corta distancia.

En la instalación del piso 2° se tienen valores de atenuación en el rango de los 20 MHz que permitirían utilizar la red eficientemente para los fines de telecomunicaciones. Ello sugiere extender el ancho de banda a frecuencias mayores y hace que el ancho de banda de 10 MHz a 30 MHz o más aparezca como atractivo para PLC indoor.

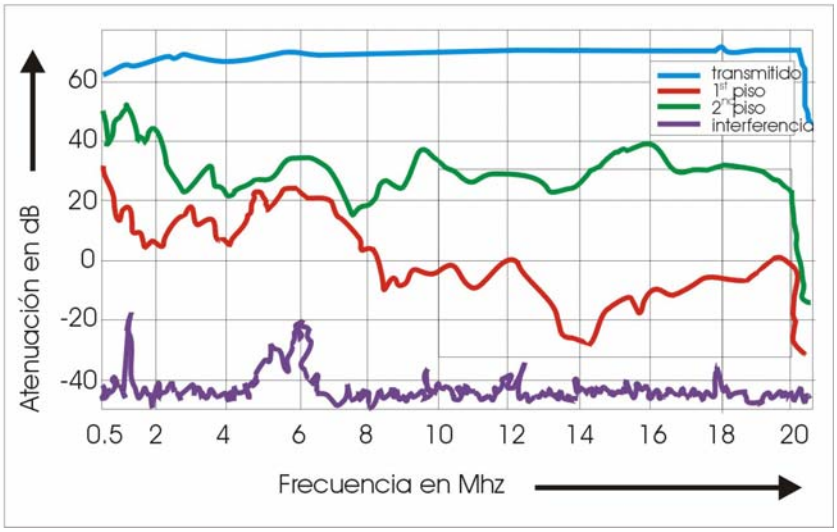


Figura 4: Curva de atenuación y ruido en una instalación domiciliaria.

Estos datos estarían de acuerdo con la norma ETSI [6] - Especificación Técnica ETSI TS 101 867 V1.1.1 (2000-11): **PowerLine Telecommunications (PLT); Coexistence of Access and In-House PowerLine Systems.** En ella, a los sistemas PLC outdoor, se los llama PLT Access; y a los indoor PLT In-House. La misma establece el ancho de banda que se destina a cada tipo de servicio y además los niveles de densidad de potencia (PSD) que deben existir para los distintos anchos de banda, y entre la separación de ellos, para la coexistencia de ambos sistemas.

Es necesario destacar que estos son establecidos para los sistemas PLC o PLT llamados de 1º fase. Se menciona que podría existir un sistema de 2º fase, que ampliaría el ancho de banda del sistema outdoor o indoor, cuando no sean utilizados simultáneamente y en lugares geográficos próximos.

Pero en este caso será necesario la existencia de un canal común de señalización, que permita por un lado la restitución del ancho de banda al sistema que le corresponda, access o in-house, y por el otro, la coexistencia de sistemas similares a través de la ***publicación*** de las particularidades de capa 1 y 2, de cada sistema; tales como modulación, acceso al medio etc.

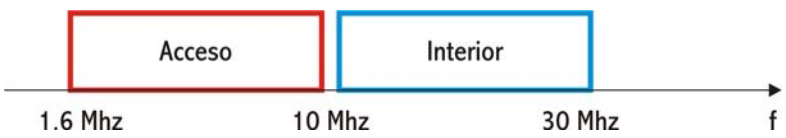


Figura 5: Ancho de banda para los sistemas PLT access e In-House

2.4 Compatibilidad Electromagnética.

Una de las mayores incertidumbres que rodean a esta tecnología, es si realmente logrará una convivencia armoniosa entre ella y las emisiones de radio amateur o radio difusión de onda corta y aún con sistemas PLC en sus cercanías.

El uso de la red eléctrica para servicios de telecomunicaciones representa una estructura electromagnética abierta e insuficientemente protegida contra la emisión o recepción de señales en alta frecuencia. De esta manera la transmisión de señales puede afectar a sistemas en redes próximas, y diferentes servicios inalámbricos vía los campos radiados.

Este fenómeno se denomina **Compatibilidad Electromagnética**. Y se refiere a la coexistencia de diferentes sistemas PLC en las proximidades; con la coexistencia a su vez, de estos con los servicios inalámbricos que se presten en la zona. La compatibilidad electromagnética de servicios de radio y sistemas PLC es básicamente bidireccional.

Esto sucede, pues a su vez, los campos electromagnéticos de los servicios de radio inducen tensiones o corrientes en la redes de distribución, dado que las líneas no malladas actúan como antenas. Por otra parte, las señales de PLC irradian campos, los cuales pueden inhibir a los radios receptores.

Mientras los sistemas PLC deben evitar las interferencias de radio, estos deben limitar la radiación en frecuencias que les han sido asignadas con anterioridad, y deben permanecer inalterables.

La Figura 6, muestra un análisis del espectro de frecuencias hasta 30 MHz. Se que solo quedan anchos de banda muy pequeños y dispersos en el espectro analizado para este servicio. No se puede tomar como definitiva esta distribución ya que hay algunos aun, que pueden usar estas frecuencias en otras parte del mundo.

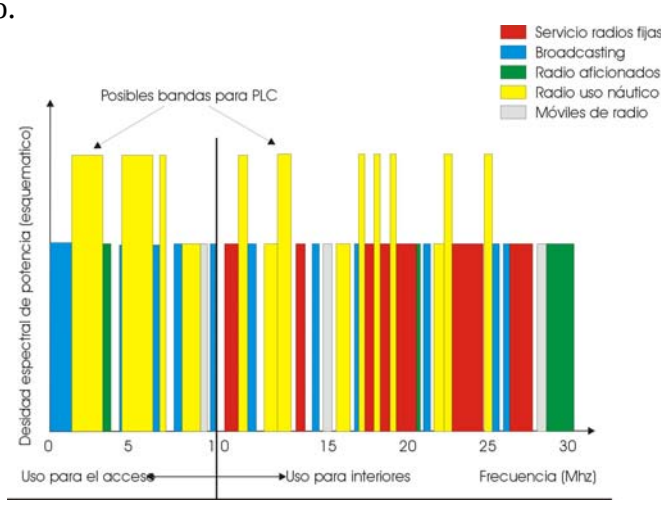


Figura 6: Servicios existentes en el espectro de frecuencias hasta 30 MHz. [1]

A futuro se debería estudiar la asignación permanente de estos intervalos de frecuencias a PLC. Ello permitiría incrementar la potencia transmitida y el uso de esquemas de modulación de multiportadoras y OFDM. De esta manera los mencionados intervalos harían más eficientes los servicios.

Otra posible forma de llevar una compatibilidad electromagnética entre PLC y los servicios inalámbricos es limitando los niveles de radiación desde los sistemas PLC.

En Alemania en 1999 el RegTP [9], a través de NB30, se sugirió una limitación en la emisión que va de 40 dBµV/m (≡ 100 µV/m) a 1 MHz cayendo a 27 dBµV/m (≡ 22,3 µV/m) a 30 MHz.

Para frecuencias por arriba de estas y hasta 1 GHz se mantiene los niveles de 27 dBµV/m. Estas limitaciones cambian actualmente de país en país. En el Reino Unido la tecnología PLC ha presentaron problemas severos de interferencias. Esto ha hecho, que por un lado la BBC haga una propuesta, y por el otro, los organismos reguladores impongan limitaciones menos severas que la anterior a través de la norma MPT 1570.

Pero estos niveles están 20 dB por debajo de la norma alemana, tanto que las redes de telefonía y redes de computadoras no cumplirían con los niveles de campos eléctrico reglamentados. Ello implicaría desalentar a la industria y a las compañías interesadas en PLC.

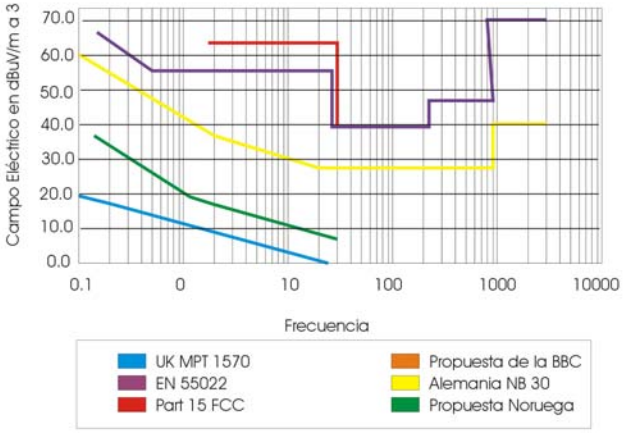


Figura 7: Emisiones de campo eléctrico para cables de comunicaciones en diferentes países.

En EE.UU. con la norma Part 15 de la Federal Communications Commission (FCC) [8], que trata de ajustarse a los estándares internacionales del International Special Committee on Radio Interference (CISPR) como el CISPR 22 [10] ha establecido niveles que están por arriba de los alemanes. Esto permitirá alcanzar mayores velocidades.

Como se puede ver en la Figura 7, hay una propuesta noruega la cual es más exigente que la inglesa. También está la norma EN 55022 de CENELEC [11] que es más permisiva que las anteriores.

De todas ellas, las especificaciones de la FCC son las más convenientes para el desarrollo de los sistemas PLC. La norma Part 15 fue modificada en el año 2000 por la FCC, por lo se interpretó como que se quería favorecer a PLC. Pero grupos de radioaficionados lograron que ella se vuelva a modificar en el año 2004 ARRL [7], aunque sin alterar los límites impuestos, sino estableciendo frecuencias prohibidas. Esto ha obligado a una certificación de los fabricantes en lugar de verificación y se da mandato para control de frecuencias, niveles de potencia y apagado si fuese necesario.

Generalmente la intensidad electromagnética decrece con el incremento de la distancia de las líneas que transportan señal. La coexistencia de equipos PLC y servicios de radio implica mantener la intensidad del campo causado por PLC tan bajo que permita recibir el servicio deseado en cualquier lugar sin interferencias. De este modo dos aspectos que también deben tenerse en cuenta para la determinación de los niveles de transmisión en PLC son el rango de frecuencia y la ubicación de los receptores de radio que posiblemente se interfiera.

3. ESTANDARIZACIÓN Y MARCO REGULATORIO.

A nivel de los sistemas de PLC, aún no se ha logrado establecer un estándar completo. Hay varias organizaciones que trabajan en la actualidad a fin de lograrlo. Las más importantes son el PLC-Forum, UPLC y PLCA [2]. A nivel de instalaciones indoor, si existe un estándar. Sobre él, las demás organizaciones están tratando de tomarlo como referencia para la normalización outdoor. Este estándar es denominado HomePlug Powerline Alliance (HPA) [5]. Si se dispusiera de múltiples estándares, se crearían mercados separados. Cada uno de ellos, produciría un reducido número de equipos y en consecuencia los precios serían muy altos, y habría muy poca competencia.

Existen dos aspectos que son los más importantes sobre los cuales se está trabajando a nivel estandarización, a fin de lograr una solución uniforme:

1) Diferencias de voltaje en los sistemas eléctricos de distintos países del mundo. Existen a nivel mundial sistemas de 110 V y 220 V, por ello se debería fijar calidades de servicio, frecuencias y comportamiento que permitan establecer una independencia del voltaje de trabajo de la red.

2) Diferencias en la regulación establecida para los niveles de emisión electromagnética. Esto sucede por cuanto los niveles permitidos de ondas irradiadas difieren en los diferentes países: ello haría que los usuarios de sistemas PLC no pueden usar un módem adquirido en otro país.

   Desde el punto de vista de los niveles de emisión, el PLC Forum y la CELENEC cooperan con el CISPR - International Special Committee on Radio Interference, quien maneja internacionalmente los niveles de interferencia de radio.

Con respecto a los aspectos regulatorios que afectan al PLC se pueden destacar:

- Regulaciones técnicas: básicamente en lo relativo a la compatibilidad electromagnética con otros servicios.
- Regulaciones de servicio: salvo requisitos técnicos, no existe ningún impedimento para proporcionar servicios de telecomunicación a través de las líneas eléctricas.

Se deberá reglamentar o no, en la Comisión Nacional de Comunicaciones lo requisitos para operar como prestador de servicios de Internet y canales de órdenes utilizando esta tecnología.

4. CONCLUSIONES.

Del estudio de todos los aspectos que deben tenerse en cuenta para la implementación exitosa de esta tecnología, se resalta como conclusiones primarias las siguientes:

- PLC puede resultar adecuado para la Argentina, que posee esquemas de diseño de redes eléctricas idénticos al europeo.
- Las ventajas que permite la sinergia de servicios, en este caso los de telecomunicaciones y el eléctrico es notable. Permite una drástica reducción de costos de prestación de tales servicios.
- No existe un avance importante en lo que hace a regulación en el uso de la tecnología a nivel mundial. En Argentina no existe absolutamente ningún avance al respecto. La CNC no tiene establecida ninguna regulación al respecto.

- Se requiere a nivel mundial, disponer de algunos aspectos en los cuales se logre normalización y el diseño de estándares entre distintos fabricantes.
- PLC deberá competir con otras tecnologías para la última milla, como ADSL y las tecnologías inalámbricas. Los costos tendrán seguramente la última palabra.

Podría agregarse como conclusiones claves a futuro, que PLC indoor ha alcanzado un desarrollo muy importante al contar con estándares internacionales como HP 1.0 de HomePlug. Éste ha permitido a las empresas desarrollar equipos compatibles que van en beneficio del usuario final. Y que PLC outdoor al no tener aun solucionado el problema de las incompatibilidades electromagnéticas con otros servicios preexistente pueden llevar al abandono de las inversiones en desarrollar tecnologías que la vinculen con los sistemas PLC indoor.

5. AGRADECIMIENTO.

Este proyecto cuenta con financiamiento de la Agencia Nacional la Promoción Científica Tecnológica, y de CITEFA - PICTO Nº 11-18621 - Préstamo BID 1728 OC-AR, a quienes agradecemos el apoyo recibido.

REFERENCIAS

[1] Doster, Klaus, T. PowerLine Communication. Prentice Hall PTR. 2000.
[2] The PowerLine Communications Association. www.plca.net.
[3] DS2, Fabricante de Circuitos Integrados. www.ds2.es.
[4] Iwan, Eduardo. AP Tecnologías de Convergencia sobre Redes de Video Cable y Redes Eléctricas. Expo Comm Argentina. Buenos Aires. 2006.
[5] HomePlug PowerLine Alliance (HPA). www.net-athome.com.
[6] European Telecommunications Standards Institute. ETSI, Francia. www.etsi.org.
[7] The National Association for Amateur Radio. http://www.arrl.org
[8] Federal Communications Commission - FCC. www.ffc.gov
[9] Federal Network Agency of Germany. http://www.regtp.de/tech_reg_tele.
[10] CISPR 22; Limits and methods of measurement of radio disturbance characteristics of information technology equipment. CISPR Publications 22. Geneva. Switzerland. 1993.
[11] EN 55022: Equipos de Tecnología de la Información. Características de las Perturbaciones Radioeléctricas. Límites y Métodos de Medida. Coincidente con CISPR 22 [10], CENELEC. Bruselas. 1998.
[12] ETSI TR 102 175 V1.1.1 (2003-03). PowerLine Telecommunications - PLT: Channel characterization and measurement methods.
[13] ETSI TR 102 324 V1.1.1 (2004-05). PowerLine Telecommunications - PLT; Radiated emissions' characteristics and measurement method of state of the art powerline communication networks.
[14] EN 55011: Límites y métodos de medida de características de perturbaciones de radio de equipos de radiofrecuencia de aplicación industrial, científica y médica (ISM).CENELEC. Bruselas. 1999.
[15] The Opera Consortium, Open PLC European Research Alliance. www.operaconsotium.com.

Desarrollo de un sistema operativo didáctico

Universidad Nacional de La Matanza - Departamento de Ingeniería e Investigaciones Tecnológicas
Florencio Varela 1603 – (1754) – San Justo – Buenos Aires – Argentina

Ing. Nicanor Casas
ncasas@unlam.edu.ar

Lic. Graciela De Luca
gdeluca@unlam.edu.ar

Sr. Martín Cortina
mcortina@unlam.edu.ar

Abstract

The main purpose of the SODIUM operating system is to allow students to compare different algorithms of processes administration, memory administration, input/output handling devices and diverse file system, using for this a parameterization at execution time. This will permit them to perform execution processes tests selecting different wanted algorithms, avoiding the making up of tedious compilations, or written comparisons, which are normally not especified in common course books because the problems solving is limited by exercises of simple input.

Other additional SODIUM purpose is to work out the algorithms in their original form, in the way they were design by their creators, and following a basic structure, bypassing in this way the characteristic adaptations of the best known operative systems, that compete for the informatic market access.

The SODIUM system also allows the different computer constitutive elements to be virtualized so that the algorithms can work and be consulted in their purest form without depending on other elements that can impair their performance.

Keywords: Kernel, Loader, Unix, FIFO, SJF, Round Robin, Priorities, FAT, FAT, IVT, BDA, DPL, GDT, PIC, TSS, RTC, IDT

Resumen

El propósito principal del sistema operativo SODIUM es el de permitir a los alumnos comparar diferentes algoritmos de administración de procesos, de administración de memoria, de manejo de dispositivos de entrada/salida y diferentes sistemas de archivos, utilizando para ello un sistema de parametrizaciones en tiempo de ejecución. Esto les permitirá realizar pruebas de ejecución de procesos seleccionando diferentes algoritmos a voluntad, evitándose tener que realizar tediosas compilaciones, o tener que realizar comparaciones escritas, que por lo general no se encuentran especificadas en los libros más comunes de la asignatura dado que la resolución de problemas está limitada a ejercicios de simple entrada.

Otro propósito del SODIUM es el de trabajar los algoritmos en su forma original, tal como fueron planteados por sus creadores, y siguiendo una estructura básica, de esta forma evitando las adaptaciones características de los sistemas operativos más conocidos, que compiten por la penetración en el mercado informático.

El SODIUM posibilita a su vez que los diferentes elementos constitutivos de la computadora puedan ser virtualizados a fin de que los algoritmos puedan funcionar y ser consultados en su forma más pura sin tener que depender de otros elementos que puedan deteriorar su rendimiento.

Palabras Claves: Núcleo, Cargador, Unix, FIFO, SJF, Round Robin, Prioridades, FAT, IVT, BDA, DPL, GDT, PIC, TSS, RTC, IDT

1. Antecedentes

Podemos dividir los antecedentes en dos tipos diferentes de actitudes que se adoptaron para la realización de este sistema operativo.

La primera fue la del estudio de los sistemas operativos actuales más comunes en el ámbito universitario como ser Windows y Linux.

La segunda es la búsqueda de otros sistemas operativos de características didácticas desarrollados en otras universidades.

1.1.- Área Windows y Linux

Con respecto a la primera el sistema operativo Windows en cualquiera de sus versiones no está abierto a los usuarios generales ni a las universidades por lo que realizar modificaciones al respecto es muy complejo y de difícil implementación para una materia que corresponde a la currícula general del plan de estudios.

En lo que respecta a Linux teníamos la ventaja de tener a nuestro alcance el código abierto del mismo y quedaba la opción de partir de un Kernel estable y probado y a partir de allí realizar las modificaciones necesarias para poder adaptarlo a nuestros requerimientos.

Después de un período importante de tiempo dedicado al estudio del mismo nos encontramos con varios problemas que pasamos a desarrollar. Si bien es sabido de que el ciclo de vida de los procesos no es el estándar, que se indican en los libros más acreditados de la materia y que se encuentran al alcance de los alumnos, se pensó que la modificación del mismo podría resultar no tan compleja, sin embargo fue todo lo contrario porque al modificar totalmente la estructura ya el resto del sistema operativo no quedó estable, por lo que se decidió comenzar desde cero con un sistema operativo propio y de paso escribir nuestras experiencias como base para el estudio de la universidad.

1.2.- Área otros sistemas operativos de estudio.

Minix: Es un clon del sistema operativo Unix distribuido junto con su código fuente y desarrollado por el profesor Andrew S. Tanenbaum [TAN97] en 1987. La última versión oficial de Minix es la 3.1.2, publicada el 8 de Mayo de 2006.

Fue creado para enseñar a sus alumnos el diseño de sistemas operativos, poseía un reducido tamaño, basado en el paradigma del micronúcleo, y amplia documentación. Apropiado para personas que desean instalar un sistema operativo compatible con Unix en su máquina personal así como aprender sobre su funcionamiento interno. Minix fue desarrollado para correr sobre IBM PC con procesador Intel 8088 o superior y se ha portado a otros sistemas

Topsy: Es un pequeño sistema operativo el cual ha sido diseñado para propósitos de enseñanza (Topsy se refiere a Teachable Operating System), este entorno de trabajo permite la práctica de ejercicios relacionados con el curso de Computación II y un ejemplo de cómo los principios básicos pueden ser implementados en un sistema operativo real.

Nachos: Es un software instructivo que permite a los alumnos estudiar y modificar un sistema operativo real. La diferencia entre Nachos y un sistema operativo real es que Nachos corre como un simple proceso Unix y simula las facilidades de generales de bajo nivel de una máquina , incluyendo interrupciones, memoria virtual y manejo de dispositivos de entrada salida..

Zeus Os: El sistema está orientado a la sustitución de las prácticas que se realizan en la asignatura de Sistemas Operativos II con el fin de no solo aprender a nivel teórico, sino también de implementar un sistema totalmente funcional. El Objetivo fue trabajar sobre núcleos de otros sistemas operativos), como puede ser KMOS, para llevar a cabo su estudio y expansión.

Minirighi: Este sistema operativo multi-threading para arquitecturas IA-32, ofrece un kernel liviano, fácil de leer por todos y permite un análisis simplificado del código fuente a diferencia de la mayoría de lo otros sistemas operativos. Autor Andrea Righi.

Ninguno de los sistemas operativos, antes nombrados, permiten la parametrización de algoritmos en tiempo de ejecución, así como tampoco muestra los detalles de la ejecución de los algoritmos y sus estadísticas.

2. Introducción

El sistema operativo SODIUM es la consecuencia del proyecto de investigación sobre sistemas operativos de características didácticas que se lleva a cabo en la Universidad Nacional de la Matanza. El mismo está desarrollado por los alumnos que cursan la materia Sistemas Operativos transfiriéndose los conceptos desarrollados, año a año, a los alumnos que continuarán la tarea dejada por su pares de años anteriores. Hasta el momento se cuenta con el arranque y la instalación del mismo en cualquier máquina que tenga como base los estándares de IA-32, así como una interfaz de línea de comandos que permite interactuar con el sistema. Con el mismo, los alumnos pueden listar en pantalla todas las tablas internas del sistema, administrar procesos, realizar volcados de memoria, lanzar pruebas de concurrencia configuradas a gusto, y obtener las estadísticas de uso de CPU y tiempos de ejecución. Los resultados de las pruebas se almacenan en un archivo que oficia de historial, y podrá ser analizado posteriormente, presentando evidencias que son base para la comprensión definitiva del comportamiento de los algoritmos estudiados.

3. Lenguaje de programación

La mayor parte del proyecto se está desarrollando utilizando lenguaje ANSI C (más específicamente el compilador GCC), pero otra parte del código se desarrolla en lenguaje ensamblador, para llegar con más facilidad a todo aquel código de más bajo nivel.

Para llegar a esta opción el equipo de trabajo consideró los siguientes parámetros:

Es necesario un lenguaje al cual se lo considere lo suficientemente sólido y flexible.

Que permita una conexión simple con ASSEMBLER para manejar sentencias de bajo nivel y que el mismo compilador permita la interacción entre los dos lenguajes.

Un lenguaje que fuese conocido por los alumnos. ANSI C responde a este requerimiento ya que es el punto de partida de las materias de programación y que por lo tanto se aplica en otras materias.

Que fuese conocido por toda la comunidad universitaria. ANSI C responde a este requisito debido a que se enseña en las más prestigiosas carreras de informática.

4.- Importancia del código escrito

Es importante tener en cuenta que el SODIUM es un sistema operativo de estudio y por lo tanto no se mide en esta primera etapa el tiempo de ejecución del mismo, por lo tanto no se realizan funciones de optimización con la periodicidad que debieran realizarse y, como hemos consignado anteriormente, dado que este desarrollo está realizado por alumnos de la facultad para cumplimentar las exigencias de una materia y que además tienen su carga laboral, puede darse el caso de que existan más de una rutina o procedimiento repetido.

4.1.- Convenciones a la escritura

Para facilitar la tarea de unificación de los trabajos e implementaciones puntuales desarrolladas por los alumnos durante el transcurso del ciclo lectivo, fue necesario establecer una notación específica para la nomenclatura de funciones y variables, aplicables durante la programación en C y assembler. Esto a su vez permite al alumno familiarizarse en poco tiempo con la totalidad del sistema. Por citar un ejemplo, las variables no signadas comienzan con una letra “u” minúscula, y luego otra letra define el tipo de dato. Una variable entera sin signo comenzaría con “ui”. También se observan casos particulares que indican el propósito de la variable, como “p” para punteros, y “h” para identificadores de ventanas. El documento completo se encuentra disponible y totalmente desarrollado en [01].

5. Niveles de protección

Generalmente todos los sistemas operativos conocidos utilizan dos niveles de protección (los cuales coinciden con el Anillo 0 y 3 de la unidad de segmentación). Tanto Linux como Windows utilizan esos dos niveles, pero en SODIUM se utilizan todos los niveles del anillo INTEL porque la función que se persigue es el conocimiento, por lo tanto todos los alumnos deberán saber como se realiza el pasaje de anillo a anillo utilizando los GATES que correspondan en el momento que corresponda.

Sabemos positivamente que el proceso será más lento pero permitirá una mayor interiorización acerca de los mecanismos provistos por la arquitectura IA-32.

6. Estructura del Kernel

Después de un largo proceso de evaluación sobre cuál sería la estructura del sistema operativo para estudio, la estructura monolítica o la estructura por capas o niveles o jerárquica, se decidió por la estructura monolítica porque es la que más rápidamente permite una toma de conocimiento y de experiencia que faltaba en el equipo de trabajo y que la estructura por capas o niveles era más compleja de transmitir a aquellos que están incursionando por primera vez en la construcción de un sistema operativo.

Es la estructura de los primeros sistemas operativos constituidos fundamentalmente por un solo programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra, es decir que tienen amplia libertad para comunicarse entre ellas.

La utilización de un procedimiento se llama directamente, no necesita mensajes, por lo que es más rápido y más fácil de conceptualizar.

Es sabido que todas las actualizaciones que se realicen en el futuro implican recompilar todo el núcleo, pero eso permite a los desarrolladores entender mejor la estructura del sistema manteniendo la configuración del Makefile.

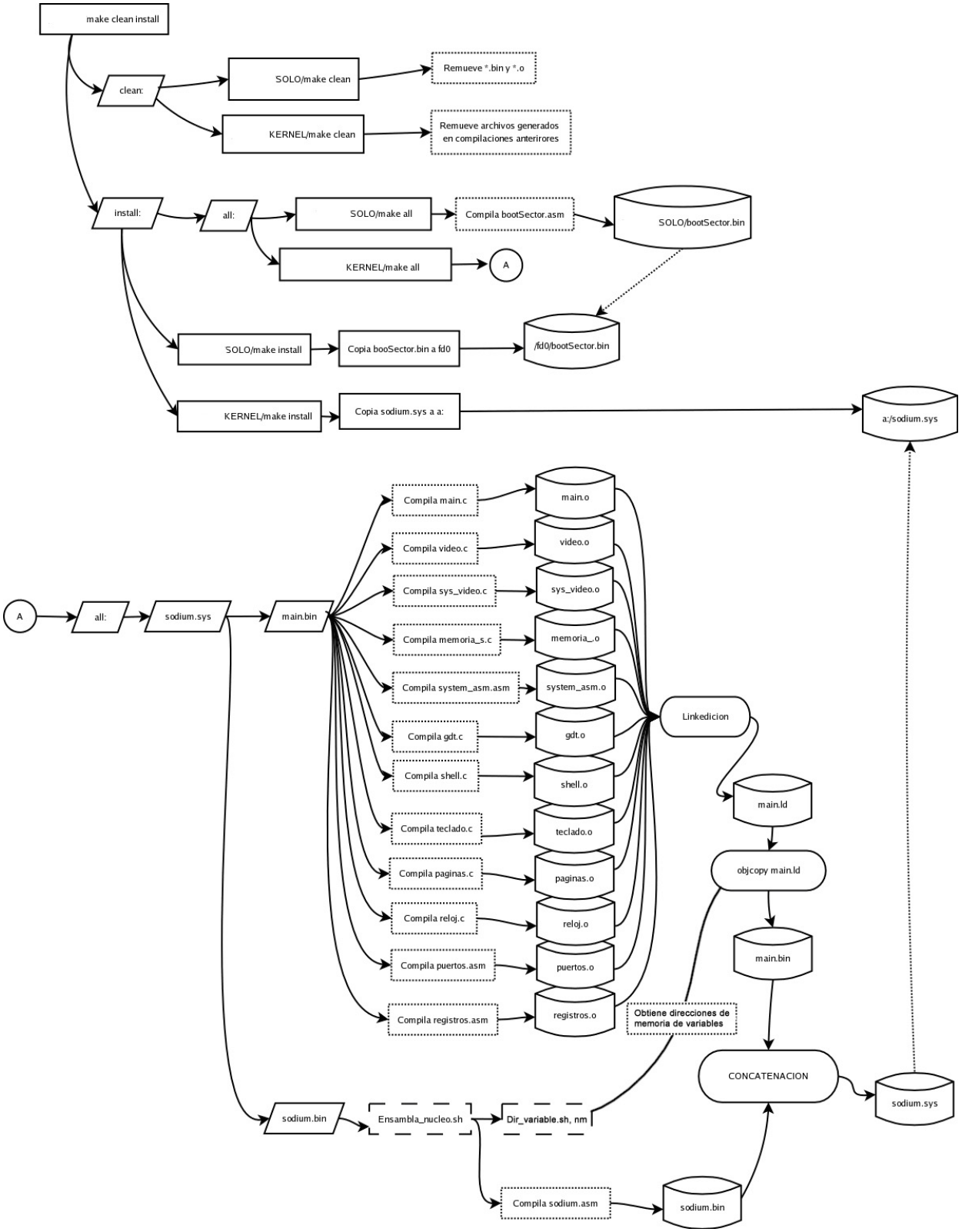
7. Plataforma de desarrollo

SODIUM se desarrolla exclusivamente con herramientas de código abierto. A continuación se detalla las versiones de los programas y utilitarios más antiguos con los que se sabe que se puede compilar el SODIUM:

Kernel	2.2.5-15	Linux RedHat 6.0, por ejemplo
Nasm	0.98.38	NetWide Assembler, compilador con soporte para notación Intel
GCC	2.91.66	Compilador C de GNU que genera archivos objeto.
DD	4.0	(GNU fileutils) Comando que copia un archivo (o stdin/out)
LD	2.9.1	Linkeditor
CAT	1.22	(GNU textutils) Concatena archivos
MAKE	3.77	Construye proyectos en base a árboles de dependencia de compilación
OBJCOPY	2.9.1	Utilidad que copia el contenido de un archivo objeto a otro tipo

También ha sido compilado correctamente en otras distribuciones más actuales, como Ubuntu, Kubuntu, Fedora, Slackware y Knoppix.

8. Diagrama de compilación, enlazado, y armado del disquete de pruebas.



9. Proceso de Carga

El SODIUM inicia ejecutando en modo real, en un entorno pre-establecido por la BIOS.

Durante la primera etapa de carga es necesario establecer el contenido de los registros de la CPU a valores conocidos, y evitar el uso de áreas de memoria antes de inicializarlas. Como es sabido, no todas las BIOS se comportan de la misma manera, de modo que durante el proceso de carga es necesario, por ejemplo, establecer mediante un salto largo los valores correctos de segmento y offset a 0x0000:(0x7C00.+ dirección_salto) para que nuestro código se ejecute correctamente.

El SODIUM ocupa relativamente poco espacio (<500kbytes), de manera que es posible ejecutarlo directamente desde un disquete, aunque también puede ser instalado y ejecutado desde un disco rígido, contando con un multi-bootloader para compatibilidad con otros sistemas previamente instalados, y los sectores de booteo correspondientes a cada medio físico. Por lo tanto, el proceso de carga varía levemente de acuerdo al marco de trabajo elegido.

Por razones de compatibilidad y simplicidad, hemos elegido, en esta primera etapa, soportar los sistemas de archivos FAT12 y FAT16. Utilizamos FAT12 en el disquete, y FAT16 en la partición del disco rígido.

Si el primer dispositivo accesible de booteo es un disco rígido, la BIOS copiará su primer sector (Master Boot Record) en la dirección lineal 0x7C00, y si el mismo contiene una firma válida (0xAA55), le cederá la ejecución. Si se trata de nuestro MBR, entonces mostrará en pantalla el listado de particiones válidas que posee el disco rígido, y nos permitirá seleccionar una de ellas mediante el teclado (eligiendo un número del 1 al 4). Nuestro MBR se reubica en la dirección lineal 0x600h y carga en la dirección 0x7C00 al primer sector correspondiente a la partición seleccionada, cediéndole luego la ejecución únicamente si posee la firma correcta (0xAA55). En nuestro caso se cede la ejecución al sector de booteo que trabaja con FAT16.

Si el primer dispositivo accesible de booteo es un disquete, la BIOS copiará su primer sector (sector 0 o sector de booteo) en la dirección lineal 0x7C00, y si el mismo contiene una firma válida, le cederá la ejecución. En nuestro caso se cede la ejecución al sector de booteo que trabaja con FAT12.

A partir de ese momento comenzamos el proceso de carga de nuestro SO, que se realizará en dos etapas.

9.1.- Primera etapa de carga (Sector de Booteo):

Consiste en establecer un stack válido, y copiar el archivo loader.sys desde el sistema de archivos del medio físico hacia la memoria. Para ello es necesario leer el directorio raíz y obtener el número de cluster inicial de dicho archivo. Luego se lee el archivo, en memoria, siguiendo cluster a cluster la cadena en la FAT (File Allocation Table) que corresponde al mismo. Cabe destacar que en un disquete de 1.44mb aloca siempre 1 sector por cluster, mientras que un disco rígido puede alocar de 2^n sectores por cluster, con n de 0 a 7. Una vez que se finaliza la carga de dicho archivo a partir de la dirección lineal 0x7e00, se le cede la ejecución, realizando un salto largo al inicio del mismo, y comienza la segunda etapa de carga.

9.2.- Segunda etapa de carga (Loader.sys)

La función del loader es la de recorrer un listado de archivos predefinido en tiempo de compilación, y copiarlos en las posiciones de memoria deseadas. Las rutinas de acceso al sistema de archivo son similares a las utilizadas durante la primera etapa, pero aquí se les da un uso más extenso y versátil.

El objetivo de esta implementación es, por un lado, continuar el proceso de carga, y por el otro, permitir a los alumnos la compilación y posterior ejecución de sus propios programas de prueba.

El listado puede verse de la siguiente manera, asumiendo dos archivos de prueba:

```
ListaDeArchivos:
db "SODIUM  SYS"    ;Nombre
db 0x12345678;Firma Inicial
db 0x87654321;Firma Final
dw 0x07E0           ;Segmento
dw 0x0000           ;Offset

db "MBR     SYS"    ;Nombre
db 0x2468ACE0;Firma Inicial
db 0x13579BDF;Firma Final
dw 0x5000           ;Segmento
dw 0x0000           ;Offset

db "PRUEBA1 SYS"    ;Nombre
db 0x11111111;Firma Inicial
db 0x22222222;Firma Final
dw 0x6000           ;Segmento
dw 0x0000           ;Offset

db "PRUEBA2 SYS"    ;Nombre
db 0x33333333;Firma Inicial
db 0x44444444;Firma Final
dw 0x7000           ;Segmento
dw 0x1000           ;Offset
FinListaDeArchivos:
```

Antes de comenzar efectivamente la carga en memoria de los archivos presentes en el listado, el loader se reubica a sí mismo a partir de la dirección lineal 0x90000. El primer archivo (sodium.sys) es parte integral del sistema operativo y siempre debe ser cargado.

El último paso del proceso de carga es el de ceder la ejecución al inicio del archivo sodium.sys.

10. Proceso de Inicialización del SODIUM

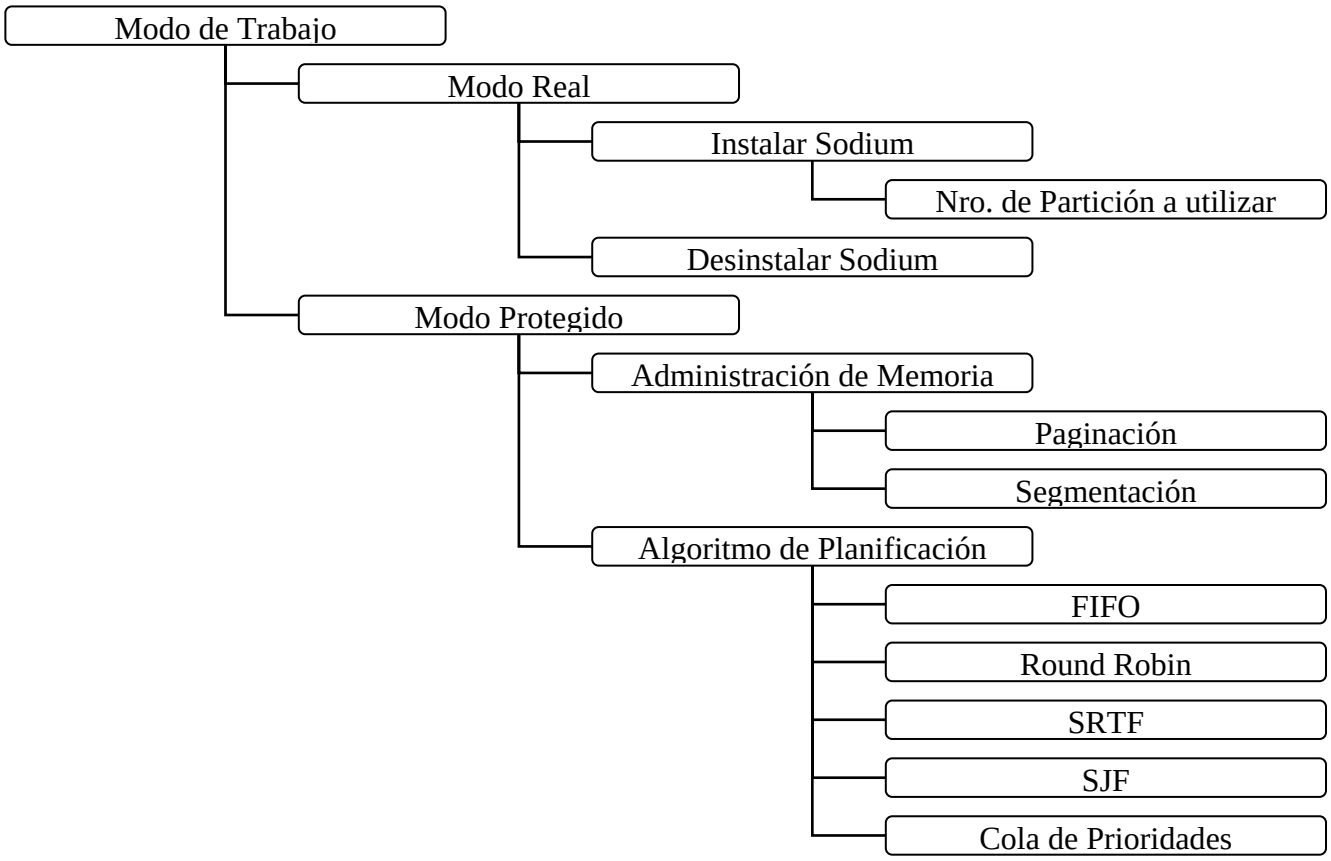
El archivo sodium.sys está compuesto por dos partes compiladas independientemente y luego concatenadas. La primer mitad es código binario ejecutable de 16bits, programado en assembler, y la segunda mitad consiste en el kernel propiamente dicho. Este último se compone exclusivamente de código de 32bits, está programado en lenguaje C y se ejecuta siempre en modo protegido. El orden de compilación de estos bloques está pensado para que durante la ejecución del código en modo real se conozca la posición de distintas variables en el kernel, y de esta forma se establezca una comunicación sencilla de las opciones de inicialización elegidas.

El proceso de inicialización también se realiza en dos etapas, culminando con la activación del planificador de tareas, que es el momento a partir del que la interfaz de línea de comandos puede leer datos del buffer.

10.1.- Primera etapa de Inicialización

Inicialmente se obtiene la cantidad de memoria base y extendida de la PC por medio de los servicios de la BIOS, y se copian dichos valores en las variables globales respectivas del kernel.

Utilizando los servicios de la bios para el manejo de pantalla y teclado, se presenta una serie de ventanas de configuración, que solicitan al usuario la elección inicial del modo de operación del SODIUM. Aquí se permite navegar por las siguientes opciones:



Una vez que el usuario está satisfecho con la configuración inicial, se copian los valores establecidos a las variables globales del kernel.

Luego se procede a habilitar la compuerta A20. Esto es necesario para poder realizar, una vez que pasemos a modo protegido, direccionamientos de más de 1MB. Dado que sabemos positivamente que no todas las computadoras aceptan el mismo mecanismo para habilitar dicha compuerta, utilizamos cuatro de los métodos más difundidos hoy en día, uno por vez, hasta verificar que el pasaje se efectuó correctamente.

Posteriormente se reubica el código de inicialización hacia una dirección más alta en memoria, cerca de los 640KB, de manera de disponer de todo el espacio que sea posible para reubicar el kernel a partir de la dirección lineal 0x0. Nótese que en este proceso estamos sobreescribiendo la IVT (Interrupt Vector Table) y la BDA (BIOS Data Area), de manera que es importante que esta copia se ejecute con las interrupciones inhibidas. De hecho, no se volverán a utilizar hasta que se termine la segunda etapa de inicialización.

Luego se inicializa a 0x0 un segmento de 64KB de memoria, contiguo a la BSS (Block Started by Symbol), que se utilizará para contener la GDT (Global Descriptor Table), y se cargan los primeros tres descriptores de la misma con el descriptor nulo, un descriptor de código, y uno de datos. Estos dos últimos, están configurados de manera que permitan direccionar 4GB a partir del inicio de la memoria, con un DPL (Descriptor Privilege Level) de 0.

Por último se establece un nuevo stack al final de la memoria convencional, y se realiza el pasaje a modo protegido seteando el bit PE del registro CR0 de la CPU y ejecutando un salto largo a la dirección lineal 0x0, usando CS (Code Selector) = 8.

10.2.- Segunda etapa de Inicialización

Aquí ya nos encontramos ejecutando en modo protegido. El proceso de inicialización ahora corresponde a la secuencia preparada en el main.c del kernel. Cada operación se confirma en la pantalla con un mensaje de “[HECHO]” u “[ERROR]”, y se muestran en pantalla las elecciones realizadas durante la etapa anterior de inicialización.

Dado que al trabajar en modo protegido necesitaremos manejar la totalidad de las excepciones de la CPU, es necesario reprogramar el PIC (Programmable Interrupt Controller) para reubicar las interrupciones de hardware a partir del índice 0x20.

Se programa el Timer, indicándole que comience a provocar interrupciones cada 0xFF ticks. Elegimos el máximo intervalo posible de tiempo entre interrupciones para que el mismo sea apreciable a la vista, una vez que el sistema está en funcionamiento.

Se inicializa la IDT (Interrupt Descriptor Table) con manejadores para las excepciones 0x0 a 0x1A, para las interrupciones de timer (0x0), teclado (0x1), disquetera y disco rígido, y por último se instala el vector de atención de pedidos de servicio (SysCalls) en la posición 0x80. Se le instruye al PIC para que desenmascare las interrupciones de estos dispositivos físicos.

De acuerdo al modo de administración de memoria se inicializa una tabla de segmentos por proceso o un mapa de bits y una tabla de directorios de página.

Luego se verifica que los archivos copiados a memoria durante el proceso de carga se encuentren en las direcciones de memoria deseadas. Para esto se busca que existan en memoria 2 firmas prefijadas de 32 bits de longitud por cada archivo, las cuales son colocadas al inicio y al fin de los mismos durante el proceso de compilación.

Se inicializa un mapa de bits que mantiene un seguimiento de las entradas libres de la GDT, para asignarlas a medida que se crean tareas, y se inicializa el vector estático de PCBs (Process Control Blocks). Posteriormente se crean tres tareas, la tarea nula, la tarea shell, y la tarea reloj. Dado que estas tres son funciones del kernel, sus respectivos selectores de código, datos y stack apuntan a los descriptores del kernel. Para cada una se crea un descriptor de tareas (Task State Segment) independiente.

- La tarea “Nula” ocupará el procesador mientras no hay otros procesos compitiendo por él.
- La tarea “Reloj” tomará la fecha y hora del RTC (Real Time Clock) y la imprimirá en pantalla.

- La tarea “Shell” se encargará de refrescar la línea de comandos con los caracteres ingresados por el teclado, manejar el historial, e interpretar los mismos una vez que se presiona la tecla “Enter”.

Por último se carga el registro TR (Task Register) con una posición de la TSS (Task State Segment) que puede utilizar para descargar el contenido de los registros del procesador al momento de ejecutar el primer cambio de contexto.

Una vez que se activa el planificador y se habilita la atención de interrupciones, el sistema se torna interactivo.

11. Presentación de la Interfaz

La interfaz de SODIUM es sencilla. Trabaja en modo terminal, con una matriz de 80x25 caracteres, y soporta una paleta de 16 colores. La escritura en pantalla se logra accediendo directamente a la memoria de video, que se encuentra mapeada a partir de la dirección 0xB8000. La pantalla se encuentra dividida en distintas áreas con propósitos específicos, como se ilustra a continuación.

11.1.- Distribución de las áreas en pantalla

<table>
<tr><td>Área de Cambios de Contexto</td><td>Área de Reloj</td></tr>
<tr><td colspan="2">Área de Impresión de Procesos Usuario</td></tr>
<tr><td colspan="2">Área de Resultado de Comandos Ejecutados</td></tr>
<tr><td colspan="2">Área de escritura de comandos</td></tr>
<tr><td colspan="2">Área de comandos útiles</td></tr>
</table>

Área de cambio de contexto: Es una línea donde el planificador escribe la información adecuada para identificar al proceso que está ejecutando actualmente. Esta es: PID (Process Id), Nombre del Programa, y el índice correspondiente en la GDT.

Área Reloj: Es una línea donde la tarea “Reloj” escribe la hora del sistema.

Área de Impresión de Procesos Usuario: Es un área de 5 líneas donde se permite que los procesos usuario impriman mensajes útiles, como el valor de contadores internos, cantidad de ciclos realizados, cantidad de operaciones E/S, etcétera. Esta ventana puede ser ocultada a voluntad, para maximizar la visualización en pantalla de los resultados de comandos ejecutados por el usuario.

Área de Resultado de Comandos Ejecutados: De acuerdo al comando ejecutado, se presenta el resultado en pantalla, que puede constar de una o cientos de líneas, como dumps de memoria, visualización de tablas del sistema, como la GDT, IDT, bitmaps, etcétera.

Área de Escritura de Comandos: Esta área puede extenderse si el comando ingresado ocupa más de una línea de longitud. Soporta dos métodos de autocompletado de comandos, por tabulación o automático por mayor similitud. Los comandos pueden recibir una cantidad variable de parámetros y los números pueden ser escritos tanto en notación decimal como hexadecimal.

También se dispone de un historial circular de hasta 20 comandos, de manera que utilizando las flechas cursoras hacia arriba o hacia abajo se puede ejecutar instrucciones tipeadas con anterioridad.

Área de Comandos Útiles: En esta única línea se listan los comandos más utilizados durante las pruebas.

12. Comandos del SODIUM

12.1.- Descripción de los comandos generales

Comando	Descripción
Ayuda	Muestra una descripción en pantalla de los comandos del SODIUM y sus parámetros.
bitmap	Muestra el mapa de memoria libre y alocada.
check [pid] [pag] [offset]	Verifica si para el proceso dado los valores de página y offset caen dentro de los límites de memoria que le fueron asignados. En caso contrario lo finaliza.
cls	Limpia la ventana de ejecución de comandos.
desc [indice]	Muestra los campos en detalle que componen al descriptor por el cual se consulta.
dump [desde] [longitud]	Realiza un vuelco de memoria en pantalla. Indicar Dirección inicial y cantidad de words a mostrar. La salida se muestra paginada.
exec [opcion] [tamaño] [prioridad]	Se crea un proceso que comparte el espacio de direcciones del kernel. Las opciones posibles son 1 y 2, siendo 1 un proceso que utiliza el 100% de CPU y 2 un proceso que imprime un string en pantalla mediante un syscall. Se indica el tamaño reservado de memoria en bytes. Esta información se utiliza para la simulación de asignación de memoria paginada y segmentada. Para los algoritmos que la utilizan, se indica una prioridad de ejecución.
execve [nombre_archivo] [parámetros] [tam_bss] [prioridad]	Se crea un proceso a partir de un archivo de prueba ejecutable compilado de forma independiente. Es necesario indicar el tamaño de BSS que resultó de dicha compilación. Para este proceso se crea un espacio de direccionamiento acotado, de modo que no puede acceder directamente a memoria de video ni a las funciones del kernel, salvo por los mecanismos de SysCalls. Los parámetros adicionales hacia el proceso se envían de la forma tradicional a través del stack. Para los algoritmos que la utilizan, se indica una prioridad de ejecución.
gdt	Muestra el contenido de las primeras 20 posiciones en la GDT.
idt	Muestra el contenido de la IDT.
kill [pid]	Remueve un proceso de la memoria. Indicar pid.
ls [dispositivo]	Muestra los archivos presentes en el directorio actual.
mem	Muestra la cantidad de Memoria base y total y disponible del sistema, y estadísticas de uso y fragmentación de la misma.
pag [pid]	Muestra los frames correspondientes a ese proceso.
ps	Lista los Procesos cargados en memoria e indica el estado de cada uno.
reboot	Efectúa un reinicio en caliente de la la PC.
segs	Muestra el nro, posición inicial y posición final de los segmentos de memoria ocupados.
set [variable] = [valor]	Se utiliza para establecer o modificar el valor de variables de entorno globales.
stack [pid] [ring 0-3]	Muestra el contenido del stack para el proceso y ring de ejecución especificado.
syskill [pid] [sig]	Envía una señal a un proceso determinado. Actualmente las señales soportadas son SIGSTOP, SIGCONT, SIGKILL, SIGTERM, SIGALRM
tss [pid]	Muestra el contenido de la TSS y PCB del proceso indicado.
ver	Indica la versión del sistema operativo.
winm	Oculta o restaura la ventana de procesos de usuario.

12.2.- Mecanismos de planificación disponibles

Comando	Descripción
planif [subcomando] [parámetros]	Bajo este comando se agrupan otros subcomandos que permiten establecer y configurar el planificador de turno.
planif aplicar	Aplica los cambios a las variables del planificador, reiniciando al mismo.
planif cambiar [nombre_algoritmo]	Establece el nuevo algoritmo de planificación a utilizar.
planif defecto	Establece los parámetros por defecto para el planificador actual.
planif mostrar	Muestra en pantalla el algoritmo utilizado actualmente y el valor de los parámetros configurables del mismo.
planif set [parametro] = [valor]	Cambia el valor de un parámetro utilizado por el algoritmo de planificación. Dichos cambios no tendrán efecto hasta que se utilice el subcomando "aplicar"

12.3.- Ejecución de pruebas por lotes

Comando	Descripción
lote [subcomando] [parámetros]	Bajo este comando se agrupan otros subcomandos que permiten administrar los Lotes de Prueba de ejecución.
lote ayuda	Se muestra la lista de comandos y su descripción.
lote cargar [nombre_archivo]	Lee un archivo descriptor de lote a memoria, y devuelve el nuevo ID
lote editar [ID]	Permite editar un lote de pruebas, seteando nuevos archivos y condiciones de ejecución, en lenguaje LCL
lote eliminar [ID]	Elimina un lote de pruebas existente.
lote guardar [ID] [nombre_archivo]	Guarda un lote de pruebas en un archivo de texto determinado.
lote iniciar [ID]	Inicia la ejecución de un lote de pruebas.
lote ls	Lista los lotes que han sido creados durante la sesión actual.
lote mostrar [ID]	Muestra las operaciones almacenadas en el lote de pruebas.
lote nuevo	Crea un nuevo lote de pruebas y muestra en pantalla su ID
lote pausar	Se detiene momentáneamente todos los procesos lanzados.
lote reanudar	Se reanuda a todos los procesos del lote de pruebas actual, si este fue pausado anteriormente.

12.4.- Extracción de los resultados de las pruebas para su posterior evaluación

Comando	Descripción
log [subcomando] [parámetros]	Bajo este comando se agrupan otros subcomandos que permiten administrar el registro.
log ayuda	Se muestra la lista de comandos y su descripción.
log eliminar	Inicializa a 0 la memoria reservada para la toma de eventos
log finalizar	Finaliza la toma de eventos.
log guardar [nombre_archivo]	Permite exportar el log a un archivo de texto con un nombre determinado en un disquete o en el disco rígido.
log iniciar	Comienza a guardar eventos en el log
log ls	Muestra el log actual en pantalla
log pausar	Se detiene momentáneamente la toma de eventos.
log reanudar	Se reanuda la toma de eventos, si esta fue pausada anteriormente.
log tamanio [Knro_eventos]	Establece el tamaño del log en múltiplos de 1024 eventos.

13. Conclusiones

De acuerdo a lo expuesto el sistema operativo SODIUM se convierte en una herramienta útil para los alumnos y profesores de las materias relacionadas. De él pueden obtenerse excelentes comparaciones para el análisis de los algoritmos más importantes. Hasta el momento sólo se relaciona con la administración de procesos, pero todo el equipo de investigación se encuentra abocado a generar la posibilidad de comparar algoritmos de administración de memoria y de administración de entrada salida generando los consecuentes drivers, manteniendo la característica de la parametrización

14. Bibliografía

[ANG98] Angulo José M. y Funke Enrique – Microprocesadores avanzados 386 y 486 – Introducción al Pentium y Pentium – Pro Editorial Paraninfo – Cuarta Edición

[BRE00] BRE00- Brey Barry B. – Los Microprocesadores Intel – Editorial Prentice Hall – Quinta Edición.

[CAR01] Card Rémy, Dumas Eric, Mével Franck - Programación Linux 2.0 API del sistema y funcionamiento del núcleo – Enrolles y Ediciones Gestión 2000 S.A.

[DEI90] Deitel Harvey M. – Introducción a los Sistemas Operativos - Addison-Wesley Iberoamericana – Segunda Edición

[INTEL] Manual de microprocesadores 386 y 486 y Pentium.

[MIL94] Milenkovic Milan – Sistemas Operativos Conceptos y diseño – Mc Graw Hill – Segunda edición

[SIL97] Silverschatz, Avi; Galvin, Peter – Operating System Concepts – Addison-Wesley Longman – Fifth Edition

[SMC00] Standard Microsystems Corporation – Application note 6.12

[STA98] Stallings Willams – Operating Systems Internals and design principles – Prentice Hall International – Third Edition

[TAN97] Tanenbaum Andrew S., Woodhull Albert S. – Operating Systems Design and Implementation – Prentice Hall – Second Edition

[TAN03] Tanenbaum Andrew S.– Sistemas Operativos Modernos – Pearson Education – SegundaEdición

[TUR03] Turley James L. – Advanced 80386 programming techniques – Osborne McGraw Hill

Internet

[01] Página de la cátedra www.souniver.com.ar

[02] INTEL 8272 Floppy disk Controller
http://andercheran.aiind.upv.es/~amstrad/docs/i8272/8272sp.htm

[03] BONAFIDE os development (detecting floppy drives)
http://www.osdever.net/tutorials/detecting_floppy_drives.php?the_id=58

[04] BONAFIDE os development (how to program the DMA)
http://www.osdever.net/tutorials/howto_dma.php?the_id=63

[05] The Unix File System
http://www.isu.edu/departments/comcom/unix/workshop/unixindex.html

[06]Cátedra de la profesora Gloria Guadalupe González Flores de la Universidad Juárez Autónoma de Tabasco trabajo sobre DMA de Roberto García García
http://mx.geocities.com/antrahxg/documentos/org_comp/procesador.html#inicio

[07] Tecnología del PC- La Placa base http://www.zator.com/Hardware/H2.htm

Entorno para el acceso remoto vía web a laboratorios multicluster

Adrián Pousa[1]**, Armando De Giusti**[2]**, Marcelo Naiouf**[3]
{apousa, degiusti, mnaiouf}@lidi.info.unlp.edu.ar

Instituto de Investigación en Informática (III-LIDI) – Facultad de Informática – UNLP

Abstract

This paper presents the research and development of an environment which allows remotely accessing multicluster Labs. The result is a management software layer that can be accessed through a Web application, allowing the remote operation of a multicluster architecture both to run programs over it and to manage it.
The Web application gives students, teachers and researches the possibility of working remotely, in a transparent fashion, assigning them a working space and allowing them to compile and run programs with different levels of interaction.
The developed software is analyzed so its use by students and researchers of concurrent, distributed and parallel processing in the UNLP.

Keywords: *Distributed systems. Clusters. Parallel Processing. Web Systems. Operating System middleware.*

Resumen

El trabajo presenta la investigación y desarrollo de un entorno que permite el acceso remoto a laboratorios multicluster. El resultado es una capa de software de administración que puede ser accedida mediante una aplicación Web, permitiendo la operación remota de una arquitectura multicluster tanto para ejecutar programas sobre ella como para administrarla.
La aplicación Web da la posibilidad a alumnos, docentes e investigadores de trabajar en forma remota sobre una arquitectura de este tipo de manera transparente, asignándoles un espacio de trabajo y permitiendo compilar y ejecutar programas con distintos niveles de interacción.
Se analiza el software desarrollado así como su utilización por estudiantes e investigadores de procesamiento concurrente, distribuido y paralelo en la UNLP.

Palabras Clave: *Sistemas Distribuidos. Clusters. Procesamiento Paralelo. Sistemas Web. Software de Sistema Operativo.*

WARSO. Workshop de Arquitectura, Redes y Sistemas Operativos

[1] Becario Telefónica e III-LIDI. Docente Auxiliar. Facultad de Informática UNLP.
[2] Investigador Principal CONICET. Profesor Titular D.E. Facultad de Informática UNLP.
[3] Profesor Titular D.E. Facultad de Informática UNLP.

1 INTRODUCCION

La utilización de redes de computadoras como maquinas paralelas virtuales se ha generalizado y el desarrollo de software de administración de recursos físicos, comunicaciones y software disponible en clusters, multiclusters y grids es un tema actual de investigación y desarrollo.

Sintéticamente, un cluster es un tipo de sistema paralelo que consiste de un conjunto de computadoras independientes, conectadas por alguna forma de red, y que configuran un recurso de procesamiento integrado [1] [2] [3]. Actualmente los clusters constituyen una clase de procesadores de alto rendimiento muy utilizados, al punto que más del 60% de las computadoras que figuran en el ranking de las "top-500" son clusters[4]. Cuando se conectan dos o más clusters sobre una red tipo LAN o WAN, se tiene alguna forma de multicluster. Existen diferentes tipos de multicluster según las características de los procesadores, el sistema operativo que utiliza cada uno y la red que los conecta [16]. Básicamente cada uno de estos aspectos define un grado de heterogeneidad en la configuración global, que será muy importante al estudiar performance [5] [17].

Un Grid es un tipo de sistema distribuido que permite compartir e integrar recursos distribuidos geográficamente, incluyendo procesadores, instrumentos, bases de datos e incluso recursos humanos. Esta clase de arquitectura permite definir múltiples configuraciones colaborativas entre comunidades de usuarios, donde cada usuario "ve" una arquitectura única [6] [7] [8].

En la Facultad de Informática existen varios subsistemas en red que son utilizados por alumnos, docentes e investigadores como maquinas paralelas virtuales: Cluster de 16 maquinas (CL1), Cluster IBM de 20 máquinas (CL2) y Cluster S101 de 16 máquinas (CL3). Además se realizan investigaciones de procesamiento paralelo empleando en conjunto todas las máquinas de CL1, CL2 y CL3. Por otra parte (dentro de los proyectos de I/D del III-LIDI [9]) estos clusters pueden trabajar "en Grid" con redes de procesadores de otras Universidades.

Los clusters están formados por distintos equipos heterogéneos en cuanto a los recursos que disponen, como cantidad de memoria y capacidad de almacenamiento y procesamiento.

El acceso a la arquitectura multicluster disponible en la Facultad requería que los usuarios trabajaran en el lugar físico donde se encuentran los clusters o bien remotamente a través de protocolos que requieren habilitar reglas de firewalls, que no dan al usuario una vista transparente y que pueden tener problemas de seguridad que se pueden evitar.

Por estas razones se decidió analizar e implementar una capa de software que permita ser accedida vía Web, brindando mayor transparencia e interacción y dando a los usuarios un espacio de trabajo sobre una arquitectura multicluster de manera que puedan compilar, ejecutar programas y obtener resultados. Por otra parte, la aplicación planteada da la posibilidad de administración remota de la arquitectura multicluster.

La capa de software se desarrolló en JAVA y se ubica por encima de los entornos de ejecución sobre arquitecturas multicluster como es el caso de los entornos MPI que proveen librerías para programación distribuida.

El trabajo está organizado como sigue: la Sección 2 define el aporte del trabajo. La Sección 3 plantea el problema de la ejecución remota. La Sección 4 plantea el esquema de solución y el entorno desarrollado, mientras la Sección 5 analiza el caso particular del acceso a los laboratorios multicluster. Finalmente, las Secciones 6 y 7 muestran algunas conclusiones y líneas de trabajo futuras respectivamente.

2 APORTE DEL TRABAJO

El aporte de este trabajo lo constituye el análisis y desarrollo de una capa de software para la administración y utilización de una arquitectura multicluster; esta capa de software puede ser accedida vía Web y brinda a los usuarios una vista transparente de la arquitectura.

3 EJECUCION REMOTA DE PROGRAMAS

La Facultad de Informática cuenta con varios clusters que son utilizados por docentes, investigadores y alumnos. El acceso a dicho equipamiento se realizaba de dos maneras: local y remotamente mediante el protocolo SSH a manera de consola de texto.

El acceso local requiere que el usuario trabaje en el lugar físico donde se encuentra el cluster, lo que restringe la disponibilidad. El acceso remoto mediante SSH si bien permite utilizar el cluster toda vez que el acceso es permitido, tiene como inconvenientes que no es transparente, que se requieren controles a nivel de firewall y se pierde seguridad.

Interesa entonces un acceso remoto, con seguridad y transparencia para los usuarios desde un entorno interactivo amigable. Descartados SSH y Telnet por las razones mencionadas en el párrafo anterior el trabajo se enfocó en el desarrollo de una capa de software accesible vía Web, que brinde al usuario una vista transparente y un espacio de trabajo para que éste opere el sistema de acuerdo con los derechos que le han sido asignados (básicamente ejecutar programas sobre el cluster y transferir los resultados).

4 SOLUCION PROPUESTA Y ENTORNO DESARROLLADO

La aplicación Web desarrollada brinda la posibilidad a alumnos, docentes e investigadores de trabajar en forma remota sobre una arquitectura multicluster de manera transparente, asignándoles un espacio de trabajo y permitiendo compilar y ejecutar programas con distintos niveles de interacción. En este trabajo se analizan las funcionalidades del software desarrollado para el acceso remoto a los Clusters de la Facultad de Informática de la UNLP, utilizados por varias cátedras de grado y postgrado, en particular las relacionadas con procesamiento concurrente, distribuido y paralelo.

El esquema de solución propuesto se muestra en la Figura 1. La capa de software (entorno) está implementada en JAVA y es accesible vía Web a través de un servidor Tomcat. Esta capa interactúa con los entornos de ejecución basados en MPI en dos de sus varias implementaciones (LAM y MPICH2).

De esta forma es posible administrar estos entornos a través de la aplicación Web, compilar programas en lenguaje C utilizando las librerías de programación distribuida provistas por los entornos MPI, además de ejecutar dichos programas y controlar las ejecuciones realizadas.

Se utiliza además un motor de base de datos MySQL para almacenar aspectos de configuración, los usuarios de la aplicación y para mantener registros sobre las ejecuciones hechas. Todas estas capas de software funcionan bajo un sistema operativo Fedora.

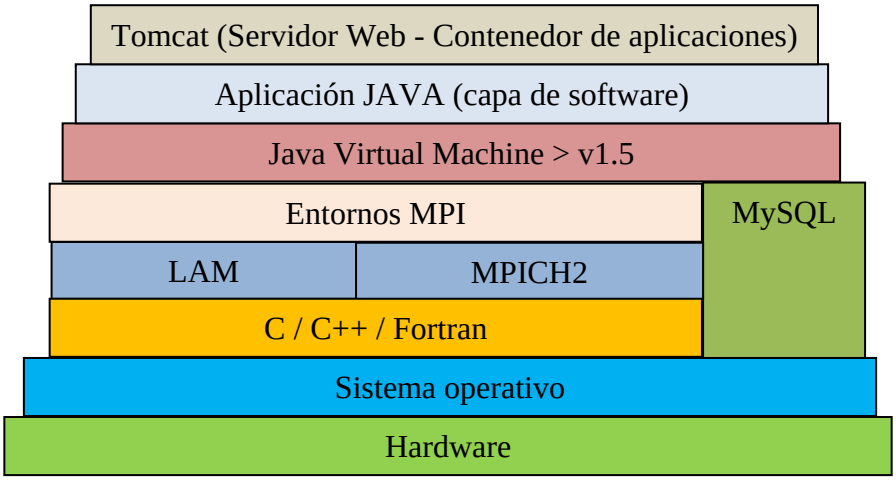


Figura 1. Esquema en capas

Es importante mencionar que todas las herramientas utilizadas tanto para el desarrollo de la capa de software como para la utilización de la arquitectura multicluster son de libre distribución y gratuitas.

En relación a la seguridad, en esta instancia se tomaron sólo medidas mínimas y es tema de trabajo futuro el caso de la seguridad en este tipo de arquitecturas.

El sistema permite ejecutar remotamente una aplicación que no se sabe que tipo de código pueda contener (esto incluye la posibilidad que exista código malicioso que pueda dañar todo el sistema).

Por lo tanto, toda la aplicación Web, incluyendo el servidor y la estructura de archivos para cada usuario, trabajan bajo un único usuario de sistema sin privilegios de administrador. Esto permite que no se puedan modificar estructuras de sistema sin permisos (pero no evita la lectura de información o el borrado de archivos sobre los que el usuario tenga permisos). En particular, por las características del sistema, cada usuario de aplicación tiene el mismo usuario de sistema y por lo tanto tiene permisos para modificar la estructura de archivos de otros usuarios de aplicación. Igualmente constituye una medida mínima para evitar daños mayores.

Toda ejecución de usuario está registrada y la aplicación Web provee un mecanismo transparente sobre la estructura de archivos que le impide obtener información extra a la que ya tiene en su espacio de trabajo, y además cada usuario está identificado lo que permite registrar cualquier operación incorrecta.

5 UTILIZACION PARA EL ACCESO A LOS LABORATORIOS MULTICLUSTER

5.1 Arquitectura

En la Facultad de Informática existen varios subsistemas de procesamiento en red que son utilizados por alumnos, docentes e investigadores como maquinas paralelas virtuales, tal como se describió en la Sección 1. Estos clusters están formados por distintos equipos heterogéneos en cuanto a los recursos que disponen como ser: cantidad de memoria, capacidad de almacenamiento y procesamiento.

Un usuario que quiera ejecutar un programa escrito en C usando la librería MPI deberá seguir los siguientes pasos:

- Conectarse remotamente vía Web a la aplicación Web escrita en JAVA
- Enviar su programa
- Compilar y pedir la ejecución del programa a la aplicación Web

Gracias a la capacidad de JAVA de ejecutar desde una aplicación escrita en este lenguaje otras aplicaciones o comandos, el usuario recibirá la salida de la ejecución de su programa C/MPI.

Desde el punto de vista académico, la solución reduce el tiempo de configuración del cluster por el alumno/usuario: simplemente se conecta a la aplicación Web y se concentra en la ejecución y evaluación de su aplicación.

5.2 Características de la aplicación Web

Cada usuario dentro de la aplicación Web tiene un espacio de trabajo semejante a un sistema de archivos de cualquier sistema operativo.

Este espacio de trabajo permite a los usuarios (con un buen nivel de interacción gracias al uso de AJAX) crear, modificar y eliminar archivos o directorios, subir archivos, comprimirlos o descomprimirlos utilizando zip, editar archivos, visualizarlos, compilar código fuente y ejecutar el código compilado en los tres modos de ejecución disponibles.

El espacio de trabajo es independiente del sistema de archivos del sistema operativo, lo que brinda un aceptable nivel de transparencia. Incluye además las tareas de administración para los usuarios administradores.

5.2.1 Perfiles de usuario

La aplicación Web permite dos perfiles de usuarios:

- Usuarios Standard (fundamentalmente alumnos): tienen un espacio de trabajo que es equivalente a un sistema de archivos; en él pueden editar, compilar y ejecutar programas en los tres modos de ejecución disponibles además de detener estas ejecuciones.

- Usuarios Administradores (docentes): agregan a las posibilidades de los usuarios comunes las tareas de administración tales como realizar altas, bajas y modificaciones de usuarios, de equipos y clusters, cambiar modos de acceso al cluster, correr o detener ambientes de ejecución MPI y manejar registros de las ejecuciones hechas.

5.2.2 Compilación y modos de ejecución

La aplicación Web permite el uso de las librerías para programación distribuida MPI utilizando el lenguaje de programación C. Es posible compilar y ejecutar programas utilizando estas librerías en dos distribuciones LAM y MPICH2.[10] [11]

Una vez compilado un programa puede ser ejecutado utilizando tres modos de ejecución:

- Sincrónico: Se envía el pedido de ejecución al servidor y se espera una respuesta en lo inmediato, por lo tanto la salida de la ejecución es retornada en la misma página WEB.

- Asincrónico: Se envía el pedido de ejecución al servidor y no se espera una respuesta en lo inmediato. Es utilizado para aplicaciones que tienen una demora importante, por lo tanto la salida de la ejecución es retornada en un archivo dentro del espacio de trabajo del usuario, y el contenido de este archivo puede visualizarse posteriormente.

- Interactivo: Da al usuario la misma sensación que tiene cuando ejecuta una aplicación de este tipo en una consola, donde la salida de la ejecución se va mostrando línea a línea. El usuario envía a ejecutar el programa y el servidor va entregando de a línea o conjunto de líneas a medida que se desarrolla la ejecución, las cuales va mostrando en la pagina Web.

Los modos sincrónico e interactivo son similares; la diferencia es que el modo sincrónico envía el pedido de ejecución y espera toda la salida de programa completa, y luego esta salida es mostrada en la página Web. Si el programa fallara por algún motivo el usuario no sabría la causa de la falla o el momento en que el programa se detuvo, recibiendo sólo una pantalla de salida en blanco o un mensaje genérico de error.

En el caso del modo interactivo, por cada línea o conjunto de líneas que retorne el programa que se este ejecutando en el cluster, se irán mostrando una por una en la pagina Web, de esta forma si el programa se detuviera o fallara inesperadamente le dará una idea al usuario de dónde puede estar el error.

Las ejecuciones sobre una arquitectura de cluster, generalmente, suelen parecerse más al modo de ejecución asincrónico donde se envía a un programa ejecutar al cluster y se espera una salida en un archivo con una demora importante de tiempo (minutos, horas, unos pocos días).

Notar que las aplicaciones de tratamiento masivo de datos, que habitualmente se ejecutan en clusters suelen tener estos requerimientos. Se supone que el programa fue anteriormente probado, asegurando la ausencia de errores, y luego ejecutado en el cluster, por lo tanto “no debería fallar”.

Los modos sincrónico e interactivo de la aplicación surgen por el uso práctico que se le da en algunas cátedras de la carrera, donde los programas no suelen tener demoras importantes, son más bien simples y el uso del cluster es con fines educativos.

5.2.3 Modos de acceso

Una arquitectura de cluster es un recurso en el que dos usuarios no deberían estar ejecutando al mismo tiempo para que puedan aprovecharse al máximo todos los recursos y tengan sentido las mediciones de tiempo de ejecución que caracterizan la perfomance en los sistemas paralelos.

La aplicación Web permite establecer dos modos de acceso:

- Exclusivo: Se asigna a un usuario una franja horaria para que pueda trabajar impidiendo que otros usuarios puedan ingresar al sistema mientras se este haciendo uso de la arquitectura. No se les permite ningún tipo de acceso a otros usuarios (ni siquiera cuando el cluster esté ocioso), y no se les permite autenticación, salvo que sean administradores y que se deba solucionar algún problema para que el usuario que tiene asignado el uso exclusivo del cluster pueda usarlo sin inconvenientes.

- Compartido: A pesar del nombre, el modo compartido, no significa que se permita el uso del cluster a dos procesos de distintos usuarios al mismo tiempo, sino que se sigue manteniendo la exclusividad en la ejecución sobre la arquitectura de cluster.
  Permite la autenticación de usuarios dejándolos operar sobre su área de trabajo, pero sólo les deja ejecutar programas si el cluster no esta ejecutando programas de otro usuario; de esta forma se aprovechan los tiempos ociosos de la arquitectura.
  En caso de que el usuario desee ejecutar un programa sobre el cluster y el mismo esté ocupado, la aplicación reintentará la ejecución hasta que logre el acceso para ejecutar o bien hasta que el usuario cancele el pedido de ejecución.

Estos dos modos surgen de las necesidades de los usuarios, los alumnos suelen correr programas de muy corta duración (algunos minutos) que no requieren un tiempo exclusivo y que bien puede compartirse con otros alumnos; en cambio, los investigadores, suelen ejecutar programas de mayor duración (horas) y requieren exclusividad de la arquitectura por un determinado periodo de tiempo.

6 CONCLUSIONES

Se ha presentado un desarrollo de entorno y soporte de administración de recursos WEB, para la utilización de arquitecturas multiprocesador tipo cluster y multicluster en forma remota, brindando una interfaz transparente.

Este tipo de solución, que integra herramientas de uso libre, es particularmente importante en las carreras de Informática, donde un número importante de alumnos de grado y postgrado requieren trabajo experimental sobre arquitecturas multiprocesador disponibles en la Facultad.

7 LINEAS DE TRABAJO FUTURAS

Existen algunos aspectos que se están estudiando para agregar mayor funcionalidad a la aplicación Web. Uno de ellos es el de ejecutar programas utilizando un manejador de recursos y un scheduler como pueden ser Torque y Maui respectivamente [12] [13]. Esta combinación permite que los programas se encolen y sean ejecutados de acuerdo a alguna política de scheduler, además de una mejor utilización de los recursos.

Otro aspecto es el de fragmentar la arquitectura multicluster de manera de tener varios clusters para distintos propósitos y con diferentes políticas. Por ejemplo, varios alumnos pueden estar utilizando parte de un cluster de forma compartida mientras que al mismo momento un investigador puede tener parte del cluster de forma exclusiva.

Así como la arquitectura puede fragmentarse puede también expandirse incorporando clusters que tal vez no estén físicamente en el mismo lugar. Actualmente, se está trabajando en la conexión de clusters de otras universidades.

Un último aspecto es el de adaptar la capa de software para que funcione en una infraestructura de grid, donde se consideran equipos como nodos de grid distribuidos ya no en una LAN (como pueden considerarse los clusters) sino en una WAN, y cada nodo puede tener a su vez clusters conectados. Esta infraestructura proporciona mayor disponibilidad de recursos y puede suponer una mayor potencia de cálculo. Un objetivo es hacer funcionar la capa de software sobre middlewares de grid como son Globus Toolkit [14] o GLite[15].

REFERENCIAS

[1] Grama A., Gupta A., Karypis G., Kumar V. “Introduction to Parallel Computing”. Second Edition. Addison Wesley, 2003.

[2] Jordan H, Alaghband G. “Fundamentals of parallel computing”. Prentice Hall, 2002.

[3] Dongarra J, Foster I, Fox G, Gropp W, Kennedy K, Torczon L, White A. “The Sourcebook of Parallel Computing”. Morgan Kauffman Publishers. Elsevier Science, 2003.

[4] The TOP500 project http://www.top500.org/

[5] Jiang, Yeung. “Scalable Inter-Cluster Communication System for Clustered Multiprocessors”. 1997.

[6] Foster I., Kesselman C., Kaufmann M. “The Grid 2: Blueprint for a New Computing Infrastructure”. The Morgan Kaufmann Series in Computer Architecture and Design. 2 edition (November 18, 2003).

[7] Juhasz Z. (Editor), Kacsuk P. (Editor), Kranzlmuller D. (Editor). “Distributed and Parallel Systems: Cluster and Grid Computing”. The International Series in Engineering and Computer Science. Springer; 1 edition (September 21, 2004)

[8] Berman F.(Editor), Fox G.(Editor), Hey A.(Editor). “Grid Computing: Making The Global Infrastructure a Reality”. John Wiley & Sons (April 8, 2003).

[9] III-LIDI http://lidi.info.unlp.edu.ar

[10] LAM/MPI Parallel Computing http://www.lam-mpi.org

[11] MPICH2/MPI Parallel Computing http://www-unix.mcs.anl.gov/mpi/mpich2/

[12] Torque Resource Manager http://www.clusterresources.com/pages/products/torque-resource-manager.php

[13] Maui Cluster Scheduler http://www.clusterresources.com/pages/products/maui-cluster-scheduler.php

[14] Globus Toolkit http://www.globus.org

[15] GLite http://glite.web.cern.ch/glite

[16] De Giusti A. et al " Parallel algorithms on Multi-Cluster Architectures using GRID Middleware. Experiences in Argentine Universities" Proceedings of the I Iberian Grid Infrastructure Conference. Spain. Mayo 2007. Pag. 322-332

[17] Naiouf M. R., De Giusti L. C., Chichizola F., De Giusti A. E. "Dynamic Load Balancing on Non-homogeneous Clusters". G.Min et al. (Eds.): ISPA 2006 Ws, LNCS 4331, pags. 65-73, 2006. Springer – Verlag. Berlin Heidelber 2006.

Estrategias de ruteo para redes móviles vinculadas a índices espacio-temporales

Daniela C. Giraudi, Gabriela S. Segura Guzmán, Edilma O. Gagliardi

Universidad Nacional de San Luis, Facultad de Ciencias Físico,

Matemáticas y Naturales, Departamento de Informática

San Luis, Argentina, D5700HHW

{dcgiraud, ggsegura, oli}@unsl.edu.ar

Resumen

Nuestro trabajo de investigación consiste en estudiar la siguiente problemática. Sobre una red móvil, total o parcialmente desconocida, transitan objetos en búsqueda de un camino que los conduzca desde un origen hacia un destino. El algoritmo de búsqueda es voraz, adaptado a utilizar diversas heurísticas definidas de acuerdo a las caracterizaciones propias de la red subyacente. Así, vinculamos las disciplinas, Bases de Datos Espacio-Temporales y Geometría Computacional. Por la primera estudiamos los modelos de red según definición del índice espacio-temporal considerado, MON-Tree y su extensión I+MON-Tree. En la segunda, estudiamos clases de grafos que modelan adecuadamente las conexiones en una red móvil, sobre las cuales se ejecutaron las estrategias de ruteo.

En este artículo presentamos las estrategias de ruteo usando el algoritmo de ruteo voraz, con heurísticas basadas en el modelo de red definido por el método de indexación espacio-temporal I+MON-Tree. Y conjuntamente, presentamos las evaluaciones experimentales realizadas para mostrar el buen desempeño de estos algoritmos de ruteo en aplicaciones espacio-temporales en redes móviles.

Palabras claves: Ruteo, grafos, redes móviles, heurísticas para algoritmos voraces de búsqueda, Geometría Computacional.

.

1. INTRODUCCIÓN

En la actualidad existe una gran variedad de situaciones en donde el problema, referenciado con el nombre de *Ruteo,* consiste en encontrar un camino entre un origen y un destino específicos, donde no se conoce en su totalidad el escenario por dónde transitar, sino que el mismo se va descubriendo a medida que se avanza. Para ello, se han desarrollado una variedad de estrategias de ruteo que, en general, tienen como objetivo encontrar el camino más corto entre dos puntos determinados en una red parcial o totalmente desconocida, utilizando nuevas clases de algoritmos de ruteo que se basan en el uso de información geográfica.

Nuestro trabajo consistió vincular dos disciplinas Bases de Datos Espacio-Temporales y Geometría Computacional, con el fin de estudiar los modelos de red propuestos en la definición de los índices espacio-temporales MON-Tree [1,2] e I+MON-Tree [7,8,9]. Luego, se consideraron algunas clases de grafos que modelan adecuadamente las conexiones en una red móvil: *Grafo de Vecindad Relativa* y *Triangulación de Delaunay*. Sobre ellos, se definieron los modelos de red propuestos por estos métodos espacio-temporales, basados en la definición de poligonales que cubren el grafo subyacente.

A partir de estos grafos y de los conocimientos parciales o totales de tales poligonales, desarrollamos las estrategias de ruteo alternativas, haciendo una adaptación del algoritmo de *Ruteo Voraz* en grafos geométricos, e incorporamos diferentes criterios de búsqueda, relacionándolos con dos heurísticas: *Distancia Euclidiana* y *Menor Angulo*.

Por lo expuesto, realizamos la evaluación experimental, aplicando los diferentes criterios de búsqueda sobre el *Ruteo Voraz*. Los resultados, en su mayoría, mostraron un incremento notable

sobre la tasa de éxito respecto del *Algoritmo Voraz* tradicional con heurísticas H_1 y H_2 en los grafos de Vecindad Relativa y Delaunay.

Este artículo está orientado a mostrar, según el escenario presentado, los desarrollos realizados en ruteo, dejando para etapas posteriores las publicaciones de los desarrollos realizados respecto de las adaptaciones de los índices espacio-temporales a redes móviles. Por tanto, está organizado de la siguiente manera. En la Sección 2 presentamos los antecedentes; en la Sección 3 presentamos nuestra propuesta; en la Sección 4 presentamos los resultados experimentales obtenidos, y en la última sección presentamos las conclusiones y visión de futuro de la investigación.

2. ANTECEDENTES

En esta sección presentamos los aspectos teóricos relevantes de los índices espacio-temporales considerados y ruteo geométrico. Para ello, presentamos las definiciones de redes basadas en el uso de poligonales que cubren una red, y los grafos geométricos usados para la modelación de las redes, conjuntamente con el algoritmo de ruteo elegido y las heurísticas planteadas.

2.1 I+MON-Tree

Este es un índice espacio-temporal que permite almacenar y recuperar información histórica y actual de objetos que transitan sobre redes fijas, basado en la versión original MON-Tree [1,2]

I+MON-Tree define dos tipos diferentes de modelos de redes. El primer modelo es *orientado a arcos*, donde la red es un grafo $G=(V, A)$ compuesto por V, un conjunto de vértices, y A, un conjunto de aristas, donde cada vértice $v \in V$ es la representación de un punto $p_v=(x, y)$ en el espacio bidimensional y donde cada arista $a \in A$ conecta dos vértices v_{1a} y v_{2a} y se le asocia una poligonal $l_a=p_1, \ldots, p_k$ donde cada p_i es un punto bidimensional $1<i<k$, donde k es el tamaño de la arista, $p_1=p_{v1}$ y $p_2=p_{v2}$. Cada objeto se mueve a lo largo de la poligonal y su posición *apos* se representa como un número entre 0 y 1, donde 0 indica que el objeto está en el vértice v_{1a} y 1 que está en v_{2a} de la arista. El segundo modelo es el *orientado a rutas*. Aquí, la red se representa como un conjunto de rutas y un conjunto de intersecciones que vinculan esas rutas, es decir, $G'=(R, I)$ donde R es el conjunto de rutas e I el conjunto de intersecciones. A cada ruta $r \in R$ se le asocia una poligonal $l_r=p_1,\ldots, p_k$, donde cada p_i es un punto bidimensional y $1<i<k$ y k es el tamaño de la ruta. Una posición *rpos* dentro de la ruta se representa como un número real entre 0 y 1, donde 0 significa que se encuentra en la posición es p_1 y 1 en la posición p_k de la ruta. Una intersección $i \in I$ está representada por dos rutas r_1 *y* r_2. En la Figura 1 se muestra un ejemplo de cada modelo de red [1]:

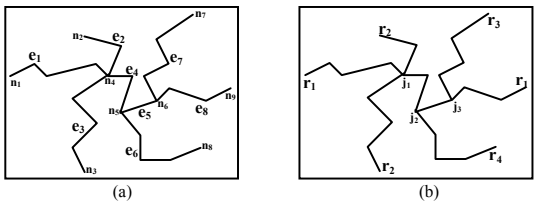


Figura 1: Ejemplo de una red (a) en el primer modelo y (b) en el segundo modelo

Considerando que este método se basa en redes fijas, nosotros extendimos las definiciones de las redes fijas a redes móviles, permitiéndonos hacer uso de las poligonales en la búsqueda de una ruta.

2.2 Ruteo Geométrico

Encontrar el camino entre dos puntos es el problema central en *Ruteo*. Para poder modelar dicho camino es necesario tener un grafo $G = (V, A)$; donde V es el conjunto de vértices que representan

objetos de un dominio específico y *A* es el conjunto de aristas del grafo, en donde cada elemento de *A* contiene un par de vértices contenidos en *V*.

Debido que la mayoría de las aplicaciones presentan restricciones en cuanto al camino que se desea encontrar, se necesita aplicar o crear una estrategia de ruteo que resulte más conveniente para cada situación particular. Con este fin surgen los *Algoritmos de Ruteo*, cuyo propósito consiste en la búsqueda de un camino que dirija un objeto desde un vértice origen a un vértice destino sobre una red.

Por esto podemos decir que, es necesario tener un grafo subyacente adecuado y un algoritmo de ruteo que permita obtener una ruta para alcanzar el vértice destino.

2.2.1 Clases de grafos

En esta subsección presentamos los grafos utilizados para modelar la red subyacente. Las redes móviles pueden ser representadas por un grafo no dirigido denominado grafo de *distancias unitarias.* Debido que las estrategias de ruteo geométrico asumen que el grafo subyacente es plano, sus aristas no se cruzan, y el grafo de *distancias unitarias* no cumple ésta condición, surge la necesidad de trabajar con subgrafos planos a partir del grafo de *distancias unitarias* tales como el grafo de *Vecindad Relativa* y *Delaunay*.

Sobre estos grafos aplicamos el algoritmo de ruteo, por lo que a continuación, presentamos las definiciones correspondientes a cada uno de ellos con sus respectivos ejemplos.

Grafo de distancias unitarias: Dado un conjunto de puntos S en el plano, el grafo UNIDAD asociado a S, UDG(S), es el grafo cuyos vértices son los puntos de S, y tal que dos vértices son adyacentes si están a distancia menor o igual a uno. [3]

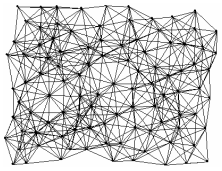


Figura 2: Ejemplo de distancias unitarias

Grafo de Vecindad Relativa: El grafo de *Vecindad Relativa* de un conjunto de puntos *V*, designado por *GVR*(*V*), es un grafo cuyos vértices son los puntos de *V* y tal que, para cada par de puntos $v,w \in V$, existe la arista (*v,w*) si ningún otro punto de *V* se encuentra en la zona de intersección de los círculos *C* y *C'*, de radio *dist*(*v,w*), con centro en los puntos *v* y *w* respectivamente. [3]

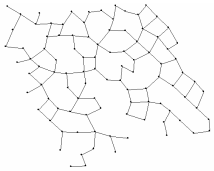


Figura 3 Ejemplo de Grafo de Vecindad Relativa

Triangulación de Delaunay: Dado un conjunto *V* de puntos del plano, una triangulación *T* de *V*, denotada por *T*(*V*), es un grafo geométrico plano cuyos vértices son los puntos de *V* y maximal respecto de la planaridad. Es decir, cualquier arista conectando dos puntos de *V* que no esté en *T*(*V*) corta a una de las existentes. [3]

Todas las caras de una triangulación son triángulos, salvo la cara no acotada. La unión de las caras acotadas de *T*(*V*) es el cierre convexo del conjunto *V*.

Figura 4: Ejemplo de Triangulación de Delaunay

Para lograr el objetivo de encontrar un camino entre un punto y otro es necesario tener en cuenta la necesidad de obtener un grafo adecuado que modele la red subyacente, los cuales fueron presentados en esta subsección. Además, es necesario utilizar un algoritmo de ruteo que dirija el camino hacia el destino deseado, los cuales presentaremos en la siguiente subsección.

2.2.2 Algoritmos de Ruteo

En esta subsección presentamos los aspectos teóricos del algoritmo de ruteo elegido y las heurísticas que se aplicaran sobre el mismo.

Ruteo Voraz (greedy routing) [13,5]: permite rutear entre un vértice origen y otro destino en redes donde la topología es desconocida o parcialmente conocida como es el caso de las redes móviles. Para cada vértice de la red selecciona cuál es el vecino inmediato que, de acuerdo a algún criterio, es el más prometedor para incluir en la búsqueda del camino.

Consideremos un grafo $G = (V, A)$, que modela la red móvil. Suponga que el vértice o desea enviar un objeto al vértice d. El vértice o sólo conoce las coordenadas de sus vecinos y del destino. El Ruteo Voraz elige el próximo vértice por el cual avanzar en la red dependiendo de una de las dos posibles heurísticas que se detallan a continuación:

Mínima distancia euclidiana (H_1): El Ruteo Voraz dirige el objeto en movimiento al vecino de o que minimiza la distancia euclidiana con el vértice d. Cuando más de un vértice se encuentra a la misma distancia hacia el vértice destino, el algoritmo selecciona uno de ellos al azar.

Menor Angulo (H_2): El Ruteo Voraz dirige el objeto en movimiento al vecino de o que minimiza el ángulo $\angle vod$. Cuando existe más de un vértice que poseen igualdad de ángulos uno de ellos se selecciona al azar. El algoritmo de Ruteo Voraz con esta heurística es también conocida con el nombre de *Ruteo por Brújula* [5, 12, 13, 4, 10].

3. PROPUESTA

En general, en Ruteo Geométrico se considera que, la elección del camino a seguir en cada vértice se basa exclusivamente en la información local almacenada en cada vértice y en la información que lleva el objeto que recorre el camino. Básicamente, la información local almacenada en cada vértice es la posición geográfica de los vértices vecinos en forma inmediata, es decir, aquellos vecinos adyacentes al vértice actual mediante alguna arista $a \in A$.

En el modelo de redes orientado a rutas y/o arcos en I+MON-Tree, se puede contar con información adicional. Ésta consiste no solo en saber cuáles son los vecinos inmediatos de un vértice, sino cuáles son las poligonales subyacentes sobre la red sobre las que se ubica el vértice. Haciendo uso del conocimiento parcial y/o total de las poligonales que representan ambos modelos, podemos ampliar el concepto de *vecinos*, de manera que no sólo recuperamos los vecinos inmediatos, sino también los vértices más alejados.

Para modelar esta situación, diseñamos diversas funciones heurísticas que orientan al algoritmo *Ruteo Voraz* a encontrar un camino entre dos puntos determinados sobre la red, con la intención de evaluar el funcionamiento del algoritmo en un entorno, total y/o parcialmente desconocido de rutas (poligonales).

Las funciones heurísticas diseñadas en este trabajo son las siguientes:

- Criterio de búsqueda por *dist-vecinos:* a partir del vértice *o* selecciona el vecino *v* que se encuentra a una distancia de longitud *dist* desde el vértice *o*, medida en vértices, que de acuerdo a la H1, es el más cercano al vértice destino *d*; o de acuerdo a la H2, es el que menos desviación presenta respecto de la línea directa que une los vértices *o* y *d*.

  En la Figura 5 vemos que, el vecino *v* es el elegido después de pasar por *dist* vecinos anteriores.

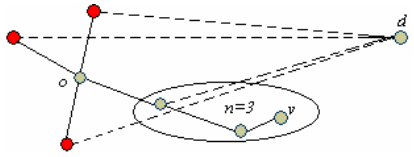


Figura 5: Criterio de búsqueda por *dist-vecinos*

- Criterio de búsqueda por *Radio r*: a partir del vértice *o* selecciona el vecino *v* que se encuentra dentro de un radio determinado de tamaño *r* y más cerca al límite del radio, que de acuerdo a la H1, es el más cercano al vértice destino *d*; o de acuerdo a la H2, es el que menos desviación presenta respecto de la línea directa que une los vértices *o* y *d*.

  En la Figura 6 vemos que, desde el origen *o* al destino *d* selecciona el vecino *v* porque es el vértice que se encuentra más cerca del límite del radio *r*.

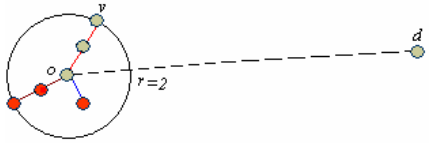


Figura 6: Criterio de búsqueda por Radio *r*

- Criterio de búsqueda por *Poligonal*: a partir de un vértice *o* selecciona aquella poligonal cuyo vértice final se encuentra más cercano al vértice destino *d*; que de acuerdo a la H1, es el más cercano al vértice destino *d*; o de acuerdo a la H2, es el que menos desviación presenta respecto de la línea directa que une los vértices *o* y *d*. Este criterio tiene la siguiente salvedad, si el vértice destino se encuentra dentro de la poligonal seleccionada, en vez de obtener la poligonal completa solo se recupera la poligonal hasta la posición del vértice destino.

  En la Figura 7 vemos que desde el origen *o* al destino *d* selecciona el vecino *v* porque es el vértice final de una de las poligonales que se encuentran más cerca del destino.

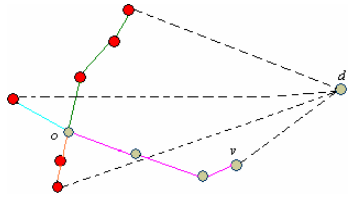


Figura 7: Criterio por poligonal utilizando *H1*

Así, nuestra propuesta consistió en diseñar diversos criterios de búsqueda combinándolos con las heurísticas tradicionales presentadas anteriormente en este artículo. Luego estas combinaciones fueron aplicadas al nuevo algoritmo de Ruteo Voraz, el cual previamente fue adaptado al modelo de

red basado en poligonales descrito por el I+MON-Tree.

4. EVALUACIÓN EXPERIMENTAL

En los experimentos realizados comparamos los resultados obtenidos por cada uno de los métodos de selección sobre un mismo grafo. Como describimos en la sección anterior, los métodos de selección son los siguientes: H_1, H_2, criterio *dist-vecinos,* criterio *radio r* y criterio *poligonal* completa. Estos tres últimos se combinan con H_1 y H_2. Trabajamos con dos tipos de grafos: Vecindad Relativa y Delaunay, en cada caso con hasta 100 nodos y la evaluación experimental aplicamos tres métricas, estas son: *Tasa de éxito*, que determina el promedio de veces que se rutea con éxito desde un vértice a otro; *Dilatación de Enlace Promedio* que determina la cantidad de vértices intermedios, en promedio, desde el vértice origen al destino; y por último, *Distancia Euclidiana Promedio* que determina la longitud promedio euclídea del camino. [2, 6]

4.1 Resultados obtenidos para grafos de Vecindad Relativa

Los resultados experimentales revelan que el criterio de búsqueda sobre el Ruteo Voraz sobre poligonales que presenta una mayor tasa de éxito es el de *dist*-vecinos con *dist*=9, seguido por el de dist-vecinos con *dist*=6 y *dist*=3. Vemos que éste criterio supera ampliamente a la heurística H_1 y H_2, en el cual nos damos cuenta que a medida que crece el *dist* hay más posibilidades de llegar con éxito al destino. Posteriormente siguen las búsquedas por radio donde ocurre lo mismo con el criterio nombrado anteriormente, a medida que el radio crece tiene mayor probabilidad de alcanzar con éxito el destino aplicando cualquiera de las dos heurísticas H_1 y H_2. Esto se debe a que, si determinamos un radio pequeño el Ruteo Voraz selecciona los vecinos que se encuentran a distancias mínimas del vértice origen, en muchos casos no existen vecinos tan cercanos en distancias a dicho vértice. El peor caso lo presenta el criterio por poligonales completas donde podemos observar que, en gran medida la tasa de éxito no supera el 38%. Esto debe a que, al seleccionar una poligonal en su totalidad, el algoritmo puede que siga un camino en el cual no hay manera de llegar al destino y por lo tanto fracasa.

Por otro lado, vemos que los resultados obtenidos para la Distancia Enlace Promedio indican que el criterio con mayor número de enlaces es el criterio *dist*-vecinos con *dist*=6 y *dist*=9. Esto se debe a que, el algoritmo siempre trata de elegir cada *dist* vecinos, provocando así una mayor cantidad de enlaces. En cuanto a los demás criterios, ellos intentan minimizar la cantidad de enlaces que deben recorrerse para llegar al destino. En este caso, muchas veces los criterios arrojan resultados similares. El criterio que presenta menor cantidad de enlaces es el criterio por poligonales completas esto se debe a que, como es un criterio que, como vimos anteriormente, no tiene una tasa de éxito elevada debido a su método de búsqueda, generalmente cuando alcanza el éxito es porque, el vértice origen y destino se encontraban cercanos, esto es, sobre la misma poligonal y con pocos enlaces entre ambos.

Finalizando con las evaluaciones experimentales para dicho tipo de grafo, podemos observar que, el mejor comportamiento lo presenta el criterio por radio. Esto se debe a que, como este criterio permite seleccionar aquellos vecinos que se encuentren a lo sumo a distancia *r* entonces el algoritmo selecciona los vecinos más cercanos a dicha distancia y en muchos casos los resultados se asemejan a la búsqueda tradicional con las heurísticas H_1 y H_2. Aparentemente, a medida que la cantidad de vértices aumenta crecen los caminos provocando una mayor distancia entre los vértices.

Podemos destacar que el comportamiento de la mayoría de los criterios de búsqueda se mantiene estable o no difiere demasiado entre las diferentes cantidades de vértices.

4.2 Resultados obtenidos para grafos de Triangulación de Delaunay

Los resultados experimentales revelan que la tasa de éxito en los grafos de Delaunay es superior a la tasa de éxito de los grafos de Vecindad Relativa. La razón de este comportamiento es la triangulación existente en el primer tipo de grafo la cual permite obtener una planarización del grafo de distancias unitarias.

El criterio *dist*-vecinos, con los diferentes *dist* y sobre las diferentes heurísticas mantiene una tasa de éxito superior al 80%. Por su parte, la búsqueda por poligonal completa a medida que aumenta la cantidad de vértices decrementa su desempeño (*performance*), esto se debe a que, cuantos más vértices hay, más largas son las poligonales que se forman y menor la probabilidad de encontrar el destino, si éste se encuentra en una poligonal distinta a la del origen. Este criterio supera los resultados obtenidos del algoritmo de Ruteo Voraz tradicional con las heurísticas H_1 y H_2.

Vemos que, como ocurría en los grafos de Vecindad Relativa el criterio por radio con r pequeño tiene menor probabilidad de alcanzar con éxito el destino. Pero a diferencia de lo que ocurría en los grafos de Vecindad Relativa las probabilidades de alcanzar el destino con éxito son mayores en los grafos de Delaunay. En la mayoría de los criterios, la disminución de la tasa de éxito se debe al incremento de vértices.

Es importante notar que los diferentes criterios no varían demasiado en sus resultados cuando se los combina con H_1 y H_2.

Por otro lado, los resultados obtenidos para la dilatación de enlace presentan características similares a las obtenidas para el caso de grafos de Vecindad Relativa. En ambos tipos de grafos no se superan los 30 enlaces. Siendo el criterio *dist*-vecinos el que mejor cumple con lo anteriormente dicho.
El criterio radio con r=5 tiene un incremento en la cantidad de enlaces en todos los casos en el grafo Delaunay con respecto al grafo de Vecindad Relativa.
Los resultados indican que, el criterio que tiene un mayor número de enlaces es el criterio *dist*-vecinos con *dist*=6 y *dist*=9. Esto se debe a que, el algoritmo siempre trata de elegir cada *dist* vecinos, provocando así una mayor cantidad de enlaces. En cuanto a los demás criterios, ellos intentan minimizar la cantidad de enlaces que deben recorrerse para alcanzar el destino. En este caso, muchas veces los criterios arrojan resultados similares.

Podemos observar que hay cambios notables en los resultados obtenidos por los criterios cuando se los combina con las heurísticas H_1 o H_2.

Para finalizar con las evaluaciones experimentales, para dicho tipo de grafo observamos que, el mejor comportamiento lo presenta tanto el criterio poligonal como el por radio, esto se debe a que, en el caso de las poligonales generalmente se encuentra el destino cuando estamos en la misma poligonal y las distancias son menores. En el caso por radio, el mismo permite seleccionar aquellos vecinos que se encuentren a lo sumo a distancia r entonces el algoritmo selecciona los vecinos más cercanos a dicha distancia y en muchos casos los resultados se asemejan a la búsqueda tradicional con heurísticas H_1 y H_2. Aparentemente, a medida que la cantidad de vértices aumenta crecen los caminos provocando una mayor distancia entre los vértices.

El criterio con mayor diferencia entre las distintas cantidades de vértices es el criterio *dist*-vecinos. Esto se debe a que, el criterio selecciona el vértice que se encuentra a *dist* vértices de distancia. Desde ese lugar, selecciona aquel que, en distancia euclidiana se encuentra más cerca del destino, pero no conocemos si los vértices intermedios seleccionados eran los de menor distancia euclidiana, siendo éste uno de los motivos de que los resultados obtenidos sean los más elevados.

Podemos destacar que, en el criterio *dist*-vecinos los resultados arrojados por las heurísticas H_1 y H_2, a medida que aumentamos la cantidad de vértices, difieren entre ellas, más que en los otros criterios.

Concluyendo, podemos destacar que al aplicar los diferentes criterios de búsqueda sobre el Ruteo Voraz se ha obtenido, en algunos casos, un incremento notable sobre la tasa de éxito del algoritmo voraz tradicional con heurísticas H_1 y H_2.

De los grafos utilizados podemos concluir lo siguiente: los grafos Delaunay son más atractivos ya que por su forma y propiedades permite tener una tasa de éxito superior.

Por otra parte, los resultados obtenidos tanto en la dilatación de enlace promedio como así también en la distancia euclidiana se asemejan en los distintos grafos.

En el Anexo Figuras mostramos los diferentes resultados obtenidos para cada uno de los criterios y sobre los diferentes grafos.

5. CONCLUSIONES Y VISIÓN DE FUTURO

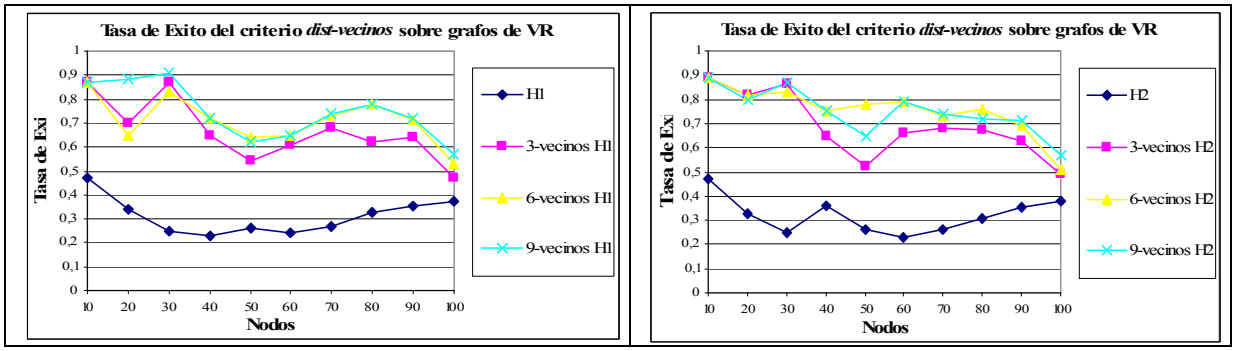
Luego de la implementación de las heurísticas presentadas en este trabajo, las evaluaciones experimentales realizadas nos brindaron resultados satisfactorios. Si bien ciertos criterios son mejores que otros, pudimos ver que, en la mayoría de los casos, la tasa de éxito lograda es superior respecto del algoritmo de *Ruteo Voraz* tradicional. Sin embargo, la Dilatación de Enlace Promedio y la Distancia Euclidiana Promedio no tuvieron grandes variaciones.

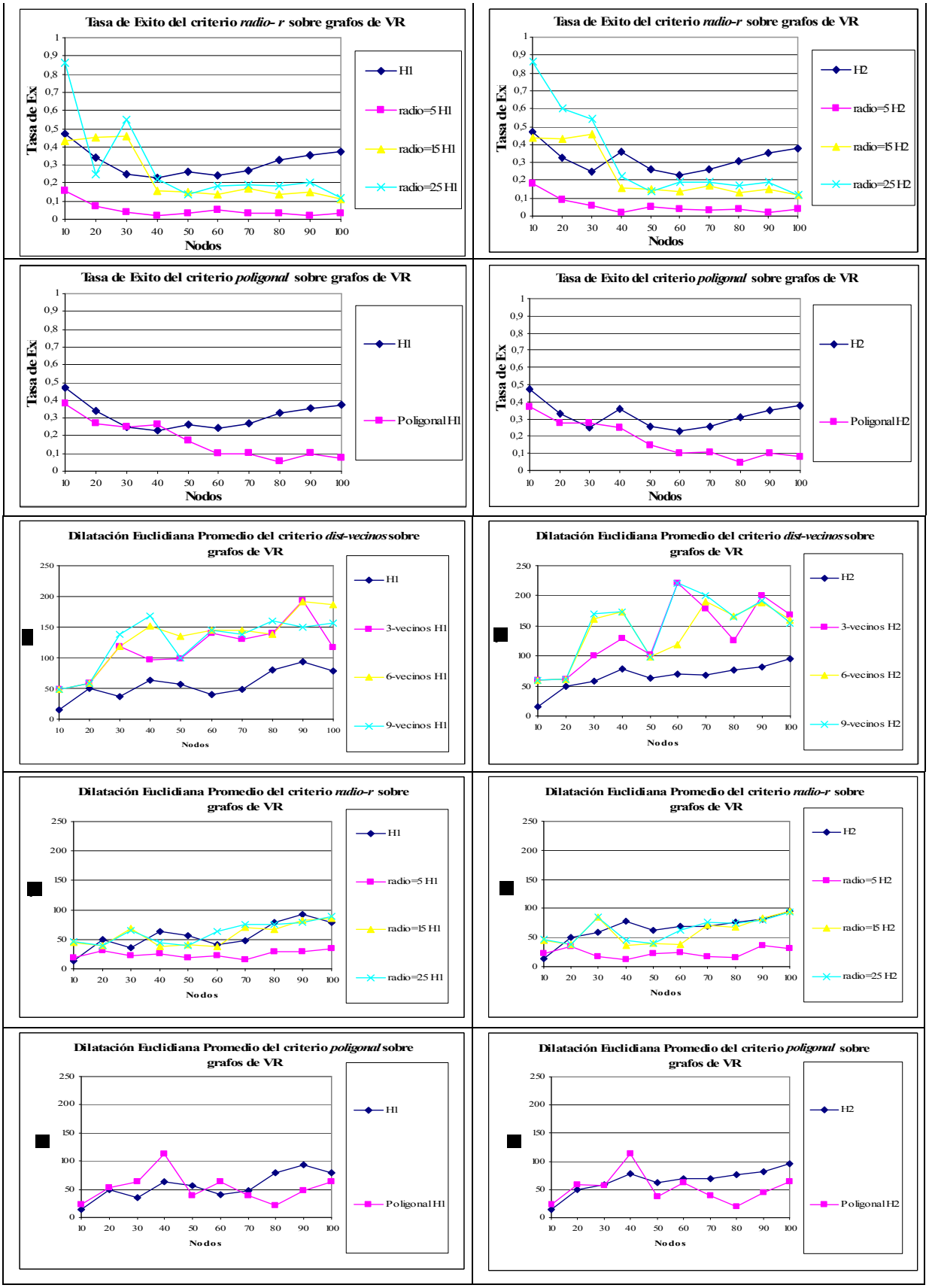
Sobre los grafos estudiados observamos que, las características de los grafos *Triangulación de Delaunay* posibilitan que los criterios de búsqueda tengan una mayor tasa de éxito. A nuestro modo de ver, éste criterio de evaluación es importante para esta ocasión porque nos permite determinar qué algoritmo de ruteo encuentra el camino entre un par de vértices con mayor frecuencia. Observamos que tanto la Dilatación de Enlace como la Distancia Euclidiana son métricas que toman importancia cuando conocemos el comportamiento de las estrategias de ruteo utilizadas.

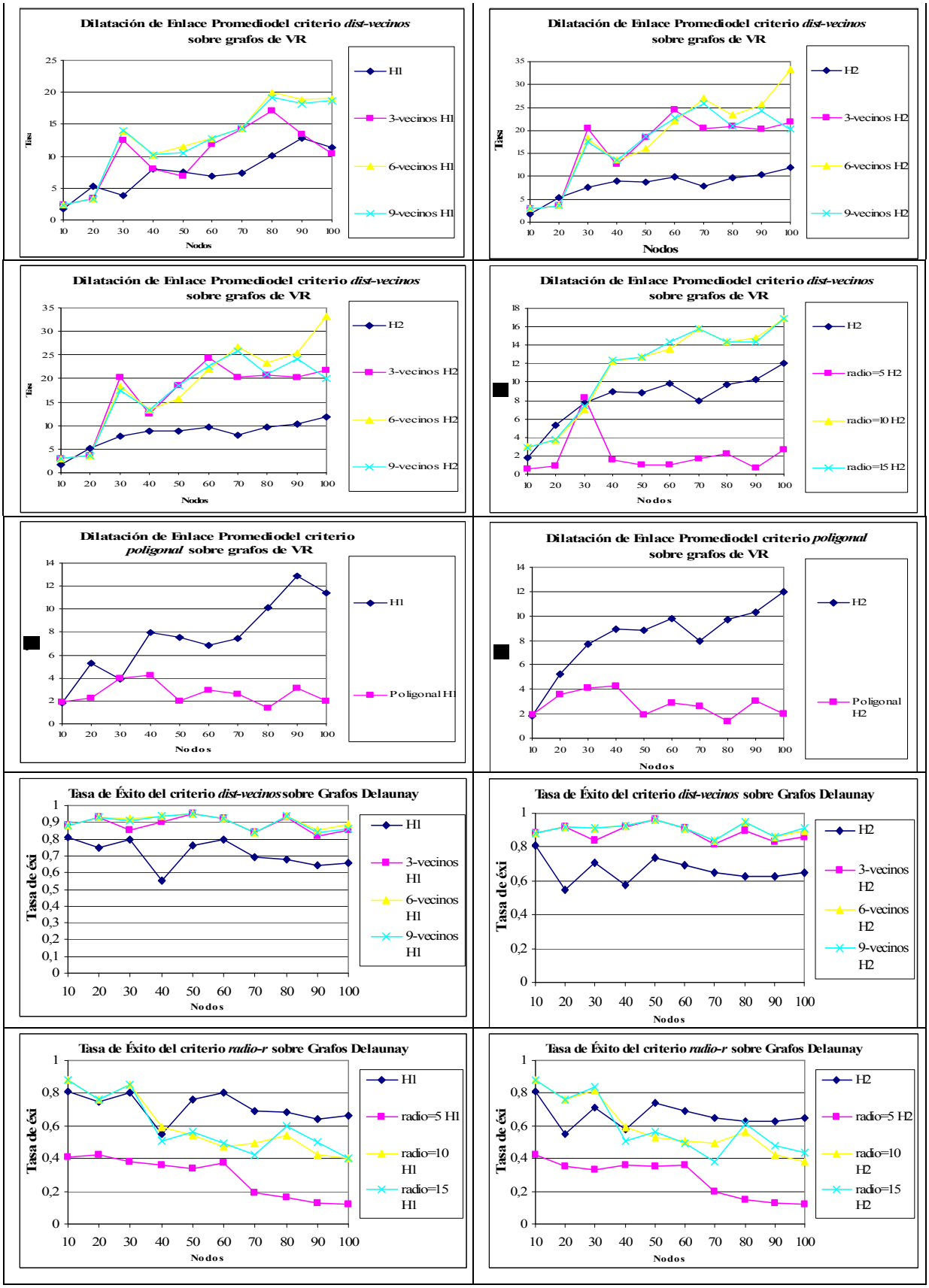
Quedan etapas por desarrollar respecto de la adaptación del método espacio-temporal a redes móviles, que si bien ya están planteados los diseños de las estructuras y algoritmos, quedan por realizar las evaluaciones experimentales [11].

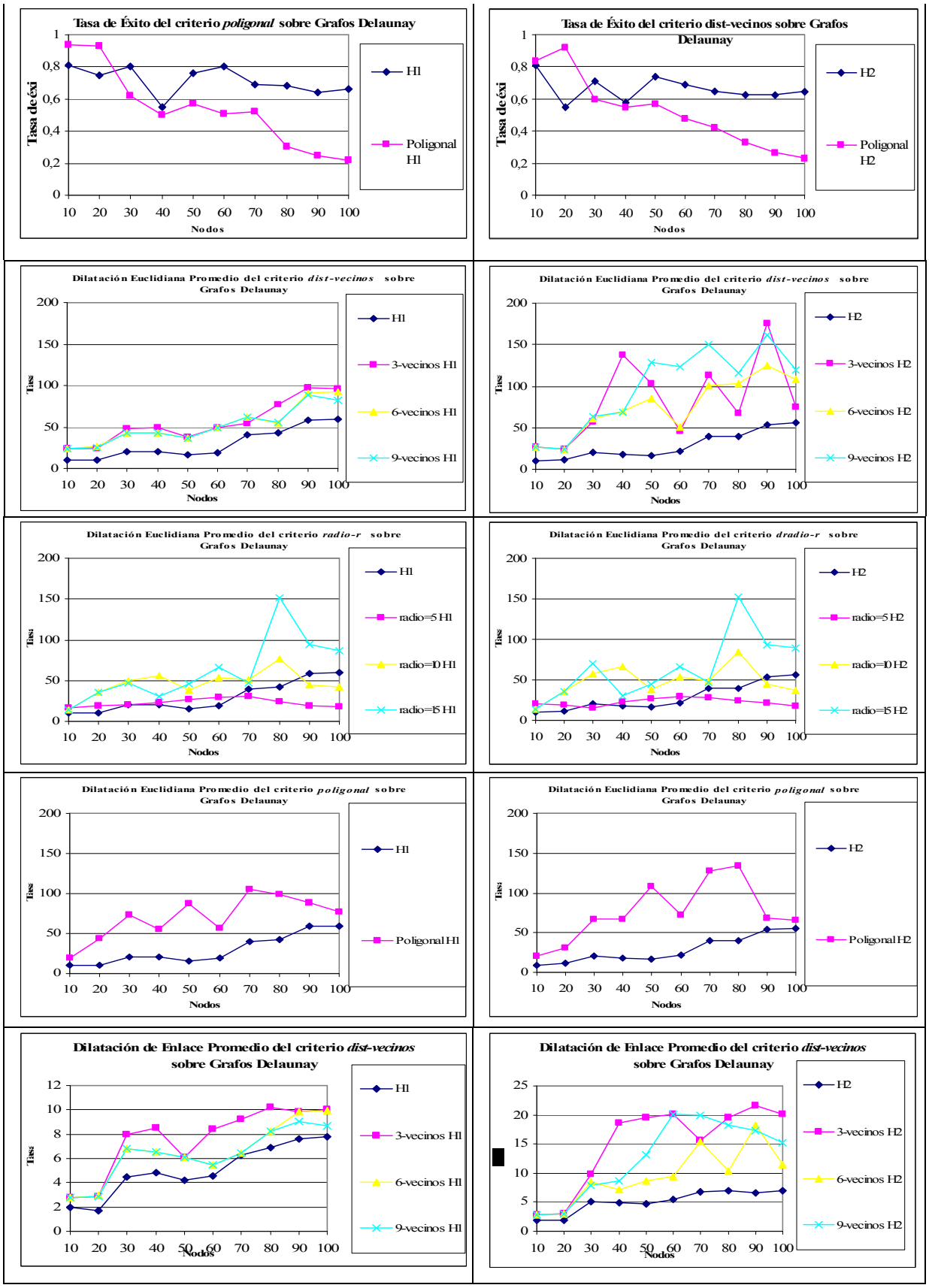
Con el desarrollo de este trabajo hemos profundizado en la temática y pretendido esencialmente destacar los resultados obtenidos, exponer los algoritmos desarrollados, consolidando y alimentando una línea de estudio, a fines de brindar un puente a investigaciones futuras.

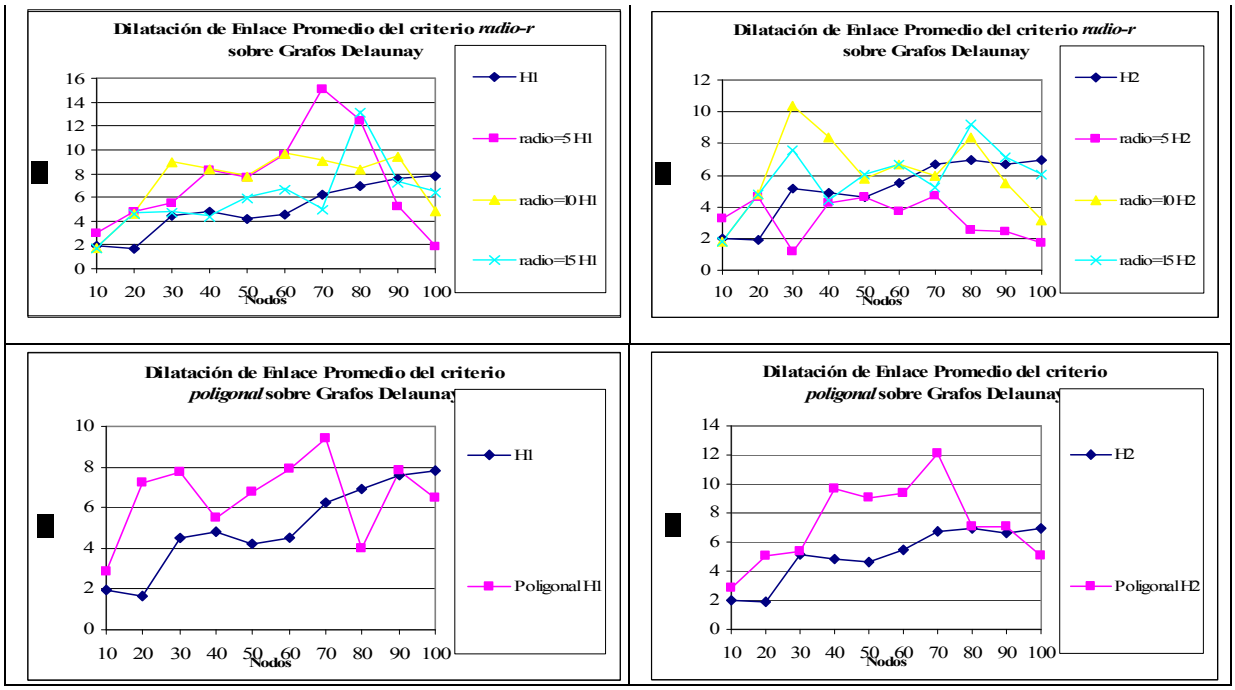
FIGURAS




Tasa de Exito del criterio radio- r sobre grafos de VR
Tasa de Ex
Nodos
H1
radio=5 H1
radio=15 H1
radio=25 H1
Tasa de Exito del criterio radio-r sobre grafos de VR
Tasa de Ex
Nodos
H2
radio=5 H2
radio=15 H2
radio=25 H2
Tasa de Exito del criterio poligonal sobre grafos de VR
Tasa de Ex
Nodos
H1
Poligonal H1
Tasa de Exito del criterio poligonal sobre grafos de VR
Tasa de Ex
Nodos
H2
Poligonal H2
Dilatación Euclidiana Promedio del criterio dist-vecinos sobre grafos de VR
Nodos
H1
3-vecinos H1
6-vecinos H1
9-vecinos H1
Dilatación Euclidiana Promedio del criterio dist-vecinos sobre grafos de VR
Nodos
H2
3-vecinos H2
6-vecinos H2
9-vecinos H2
Dilatación Euclidiana Promedio del criterio radio-r sobre grafos de VR
Nodos
H1
radio=5 H1
radio=15 H1
radio=25 H1
Dilatación Euclidiana Promedio del criterio radio-r sobre grafos de VR
Nodos
H2
radio=5 H2
radio=15 H2
radio=25 H2
Dilatación Euclidiana Promedio del criterio poligonal sobre grafos de VR
Nodos
H1
Poligonal H1
Dilatación Euclidiana Promedio del criterio poligonal sobre grafos de VR
Nodos
H2
Poligonal H2


Dilatación de Enlace Promediodel criterio dist-vecinos sobre grafos de VR
Tasa
Nodos
H1
3-vecinos H1
6-vecinos H1
9-vecinos H1
Dilatación de Enlace Promediodel criterio dist-vecinos sobre grafos de VR
Tasa
Nodos
H2
3-vecinos H2
6-vecinos H2
9-vecinos H2
Dilatación de Enlace Promediodel criterio dist-vecinos sobre grafos de VR
Tasa
Nodos
H2
3-vecinos H2
6-vecinos H2
9-vecinos H2
Dilatación de Enlace Promediodel criterio dist-vecinos sobre grafos de VR
Nodos
H2
radio=5 H2
radio=10 H2
radio=15 H2
Dilatación de Enlace Promediodel criterio poligonal sobre grafos de VR
Nodos
H1
Poligonal H1
Dilatación de Enlace Promediodel criterio poligonal sobre grafos de VR
Nodos
H2
Poligonal H2
Tasa de Éxito del criterio dist-vecinos sobre Grafos Delaunay
Tasa de éxi
Nodos
H1
3-vecinos H1
6-vecinos H1
9-vecinos H1
Tasa de Éxito del criterio dist-vecinos sobre Grafos Delaunay
Tasa de éxi
Nodos
H2
3-vecinos H2
6-vecinos H2
9-vecinos H2
Tasa de Éxito del criterio radio-r sobre Grafos Delaunay
Tasa de éxi
Nodos
H1
radio=5 H1
radio=10 H1
radio=15 H1
Tasa de Éxito del criterio radio-r sobre Grafos Delaunay
Tasa de éxi
Nodos
H2
radio=5 H2
radio=10 H2
radio=15 H2


Tasa de Éxito del criterio poligonal sobre Grafos Delaunay
Tasa de éxi
Nodos
H1
Poligonal H1
Tasa de Éxito del criterio dist-vecinos sobre Grafos Delaunay
H2
Poligonal H2
Dilatación Euclidiana Promedio del criterio dist-vecinos sobre Grafos Delaunay
3-vecinos H1
6-vecinos H1
9-vecinos H1
3-vecinos H2
6-vecinos H2
9-vecinos H2
Dilatación Euclidiana Promedio del criterio radio-r sobre Grafos Delaunay
radio=5 H1
radio=10 H1
radio=15 H1
Dilatación Euclidiana Promedio del criterio dradio-r sobre Grafos Delaunay
radio=5 H2
radio=10 H2
radio=15 H2
Dilatación Euclidiana Promedio del criterio poligonal sobre Grafos Delaunay
Poligonal H1
Poligonal H2
Dilatación de Enlace Promedio del criterio dist-vecinos sobre Grafos Delaunay



REFERENCIAS

[1] Almeida, V. y Güting, R. Indexing the trajectories of moving objects in networks. In *Proc. of the 16th Intl. Conf. on Scientific and Statistical Database Management (SSDBM)*. 2004.

[2] Almeida, V. y Güting, R. Indexing the trajectories of moving objects in networks (Extended Abstract). In *Proc. of the 16th Intl. Conf. on Scientific and Statistical Database Management (SSDBM)*. 2004

[3] Berón, M. Un evaluador de algoritmos de ruteo. Trabajo final de postgrado, Universidad Nacional de San Luis, 2005

[4] Beron, M.; Flores, S.; Gagliardi, O. *Ruteo con Brújula en Redes sin Cables*. Argentina. CACIC 2001. Congreso Argentino de Ciencias de la Computación. 2001.

[5] Bose, P.; Morin, P. *Online Routing in Triangulations*. In Proceedings of the Tenth International Symposium on Algorithms and Computation (ISAAC'99), volume 1741 of Springer LNCS, pages 113-122, 1999.

[6] Berón, M; Gagliardi, O; Peñalver, G. *Estrategias de ruteo alternativas para redes móviles.* Argentina **.** CACIC 2005. Congreso Argentino de Ciencias de la Computación. 2005.

[7] Correa. L, Ortiz. N, I + MON-Tree: índice espacio-temporal para objetos en movimiento; Trabajo final de la Licenciatura en Ciencias de la Computación, *Univ. Nac. de San Luis, Argentina,* 2006. Gagliardi, O., Directora.

[8] Correa, M.L.; Ortiz, N.J.; Gagliardi, E.O.; Gutierrez, G.. *Un método de acceso espacio-temporal restringido a redes fijas*. Workshop de Investigadores en Ciencias de la Computación 2006, WICC 2006. La Matanza, Mayo de 2006.

[9] Correa, M.L.; Ortiz, N.J.; Gagliardi, E. O.*; I+Mon-Tree: Índice Espacio-Temporal para Objetos en Movimiento.* XII Congreso Argentino de Ciencias de la Computación 2006 (CACIC 2006). San Luis, 17-21 de Octubre de 2006.

[10] Gagliardi, O.; Taranilla, M.; Beron, M. *La Geometría Computacional a nuestro alrededor.* III Workshop de Informática y Ciencias de la Computación. Argentina. 2002.

[11] Giraudi, D; Segura, G; Gagliardi, O.; Peñalver, G. Beron, M. Avances en el estudio de BDET y Ruteo sobre redes móviles**.** IX Workshop de Informática y Ciencias de la Computación. Argentina.2007.

[12] Kranakis E.; Singh H.; J. Urrutia. *Compass rotuing on Geometric network.* Proc. 11th Canadian Conference on Computational Geometry, Vancouver. 1999.

[13] *Morin, P.* Online Routing in Geometric Graph*. Tesis Doctoral. Canada. 2001.*

FPGA-Based Digital Filters Using Bit-Serial Arithmetic

Mónica Arroyuelo
Jorge Arroyuelo
Alejandro Grosso
Departamento de Informatica
Universidad Nacional de San Luis
Republica Argentina
{mdarroyu,bjarroyu,agrosso}@unsl.edu.ar

Abstract

This paper presents an efficient method for implementation of digital filters targeted FPGA architectures. The traditional approach is based on application of general purpose multipliers. However, multipliers implemented in FPGA architectures do not allow to construct economic Digital Filters. For this reason, multipliers are replaced by Lookup Tables and Adder-Substractor, which use Bit-Serial Arithmetic. Lookup Tables can be of considerable size in high order filters, thus interconnection techniques will be used to construct high order filters from a set of low order filters. The paper presents several examples confirming that these techniques allow a reduction in logic cells utilization of filters implementation based on Bit-Serial Arithmetic concept.

Keywords: Digital Filter, FIR-Filter, FPGA, IIR-Filter, Lookup Tables.

1 INTRODUCTION

A Digital Filter is a Linear Time Invariant (LTI) system, which performs numerical calculations on sampled values of the signal. The analog input signal must first be sampled and digitized using an Analog to Digital Converter (ADC). The resulting binary numbers, representing successive sampled values of the input signal, are transferred to the filter, which carries out numerical calculations on them. These calculations typically involve multiplying the input values by constants and adding the products together. If necessary, the results of these calculations, which now represent sampled values of the filtered signal, are output through a Digital to Analog Converter (DAC) to convert the signal back to analog form. In the last years digital filters have been recognized as primary digital signal processing (DSP) operation.

There are two basic types of digital filters, Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. FIR and IIR filters are used in many digital signal processing systems to perform a variety of signal filtering and conditioning functions. An IIR filter is capable of emulating the transfer functions of analog continuous-time filters, such as low-pass, band-pass, high-pass, and all-pass (phase-shifting) types of filtering. IIR filters exhibit similar phase characteristics as their analog counterparts. For arbitrary transfer functions with linear-phase response, FIR filters are utilized and have no equivalent in the analog domain.

On the other hand, the advances in Field Programmable Gate Arrays (FPGA) technology have enabled these devices to be applied to a variety of applications traditionally reserved for Application Specific Integrated Circuits (ASICs). The advantages of the FPGA approach to digital filter implementation include: higher samples rates than those that are available form traditional DSP chips, lower costs than an ASIC for moderate volume applications, and are more flexible than the alternate approaches.

A filtering function is usually carried out by a number of multiplication operations, which are expensive in terms of time and space. Therefore, several techniques are used to minimize the hardware needed to implement a filter. A technique widely used is to replace Bit-Parallel by Bit-Serial structures.

Bit-Parallel structures process all the bits of input data simultaneously at a significant hardware cost. Bit-Serial, by comparison, process the input one bit at a time. The advantage of the last one is that all the bits pass through the same logic, resulting in a huge reduction in the required hardware. Typically, the Bit-Serial approach requires $1/n^{th}$ of the hardware required for the equivalent n-bit parallel design. The price of this logic reduction is that serial hardware take n clock cycles to execute, while the equivalent parallel structure executes in one clock cycle. Since for certain classes of applications, FPGA utilization is high, performance goals are achieved while using economically attractive FPGA devices. For applications that require high speed performance, Bit-Parallel structures yields the highest performance.

This paper illustrates a new approach to the design of digital filters using Bit-Serial Arithmetic, which will reduce the logic cells utilization in an FPGA considerably, it allow us to construct high order filters (FIR-filters require a large number of coefficients to produce adequate frequency response, so these filters can occupy all the FPGA), or have others applications running on our FPGA simultaneously. Although this approach degrades the performance of filters, this degradation is not considerable for the practical purposes since the most applications do not require high speed performance. Others approaches can be see in [1],[2],[3],[4] and [5], which keep high performance but do not reduce the logic cells utilization significantly due to the fact that these try a balance between time and space.

2 IIR-DIGITAL FILTERS

IIR-Digital Filters are widely used in digital signal processing applications. They compute an output from a set of input samples and a set of previous outputs, which are multiplied by a set of coefficients and then added together to produce the output. The digital filter behaviour is determined by the filter coefficients. A general IIR-filter is characterized by the following equation:

$$y^n = a_0 x^n + a_1 x^{n-1} + \cdots + a_p x^{n-p} + b_1 y^{n-1} + \cdots + b_p y^{n-p} \quad (1)$$

where p is the filter order, the a_p's and b_p's are coefficients, x^n is the filter input at the time step n, and y^n is the filter output at the time step n.

Expanding the equation 1 for y^n in terms of the individual bits for the two-complements (2'C) operands $x = (x_{(0)}.x_{(-1)}x_{(-2)}....x_{(-l)})_2$ and $y = (y_{(0)}.y_{(-1)}y_{(-2)}....y_{(-l)})_2$ we get [6]:

$$\begin{aligned} y^n =& a_0\left(- x^n_{(0)} + \sum_{j=-l}^{-1} 2^j x^n_{(j)} \right) + a_1\left(- x^{n-1}_{(0)} + \sum_{j=-l}^{-1} 2^j x^{n-1}_{(j)} \right) + \cdots + a_p\left(- x^{n-p}_{(0)} + \sum_{j=-l}^{-1} 2^j x^{n-p}_{(j)} \right) \\ &+ b_1\left(- y^{n-1}_{(0)} + \sum_{j=-l}^{-1} 2^j y^{n-1}_{(j)} \right) + \cdots + b_p\left(- x^{n-p}_{(0)} + \sum_{j=-l}^{-1} 2^j y^{n-p}_{(j)} \right) \end{aligned} \tag{2}$$

Define $f(s,t,\ldots,u,v,\ldots,w) = a_0 s + a_1 t + \cdots + a_p u + b_0 v + \cdots + b_p w$, where s,t,. . .,u,v,. . ., and w are single-bit variables. If the coefficients are m-bits constants, then each of the 2^{2p+1} possible values for f is representable in $(m + \lceil \log_2 (2p+1) \rceil)$ bits, as it is the sum of $(2p+1)$ m-bit operands. These values can be precomputed and stored in a $((2^{2p+1}) \times \ (m + \lceil \log_2 (2p+1) \rceil))$-bit table.

Using the function f, we can rewrite the expression for y^n of the equation 2 as follows:

$$y^n = \left(\sum_{j=-l}^{-1} 2^j f(x^n_{(j)}, x^{n-1}_{(j)}, \ldots, x^{n-p}_{(j)}, y^{n-1}_{(j)}, \ldots, y^{n-p}_{(j)}) \right) - f(x^n_{(0)}, x^{n-1}_{(0)}, \ldots, x^{n-p}_{(0)}, y^{n-1}_{(0)}, \ldots, y^{n-p}_{(0)}) \tag{3}$$

Figure 1 shows the filter architecture (using Bit-Serial Arithmetic) to compute the equation 3, where the mapping f is presented as a Lookup Table (LUT) that includes all the possible linear combinations of the filter coefficients, as was mentioned previously.

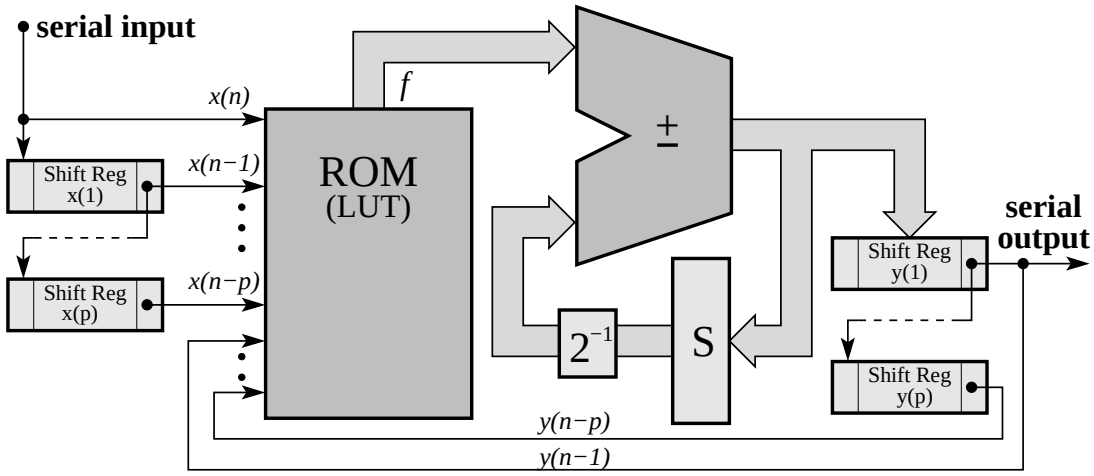


Figure 1: IIR-Digital Filter Architecture.

The architecture shown in Figure 1 has bit serial input and output. The ROM memory is addressed by the Least Significant Bits (LSB) of the x's and y's shift registers, and its output together with the S register value are fed to the *adder-substractor* where are processed. Then the *adder-substractor* result is accumulated in the S register again. After $l+1$ cycles the obtained value is the filter output, which is stored into the $y(1)$ shift register for future computations. Then, the S register is reset in 0 and a new accumulation cycle begins.

We can construct high order filters by using the previously mentioned method, but the size required for the LUTs will grow exponentially with the number of filter coefficients. For this reason, a scheme is shown to construct high order IIR-filters making use of the properties of LTI systems such as *association* and *commutation*. The associative property means that we may analyze a complicated LTI system by breaking it down into a number of simpler subsystems. The commutative property of LTI systems means that if subsystems are arranged in series, or cascade, then they can be rearranged in any order without affecting overall performance [7]. Therefore, interconnecting low order sub-filters appropriately we can make high order filters. This technique permits us to use a set of smaller LUTs instead one huge LUT, which reduces considerably the space occupied in an FPGA. Figure 2 shows the interconnection scheme, where the input, the output and the internal connections (between the filters) are serials, and the (i)-filter output is connect to the $(i+1)$-filter input straight forward.

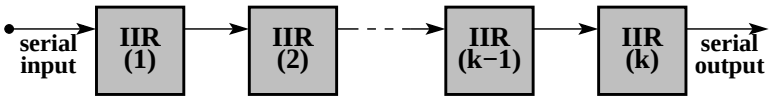


Figure 2: High Order IIR-Filter Interconnection Scheme.

For example, if we need to build a fifth-order IIR-filter we can use two second-order IIR-filters and one first-order IIR-filter. This allows us to use two 32-entry tables and one 8-entry table instead of one 2048-entry table.

3 FIR-DIGITAL FILTERS

In a FIR-Digital Filter the output depends only of present and previous input samples, which are multiplied by a set of coefficients and then added together to produce the output. The filter behaviour is determined by the filter coefficients. A general FIR-filter is characterized by the following equation:

$$y^n = a_0 x^n + a_1 x^{n-1} + \cdots + a_q x^{n-p} \tag{4}$$

Where p is the filter order, the a_p's are the filter coefficients, x^n is the input signal at the time step n, and y^n is the output signal at the time step n. The major disadvantage of these filters is that usually a large number of coefficients are required to control adequately their frequency response. Practical FIR-Filters typically need between 10 and 150 coefficients. This make them slower in operation than most IIR-filter design.

Expanding the equation 4 for y^n in terms of the individual bits for the 2'C operands $x = (x_{(0)}.x_{(-1)}\ x_{(-2)}....x_{(-l)})_2$ and $y = (y_{(0)}.y_{(-1)}y_{(-2)}....y_{(-l)})_2$, like it was made for IIR-filter, we get:

$$y^n = \left(\sum_{j=-l}^{-1} 2^j f(x^n_{(j)}, x^{n-1}_{(j)}, \ldots, x^{n-p}_{(j)}) \right) - f(x^n_{(0)}, x^{n-1}_{(0)}, \ldots, x^{n-p}_{(0)}) \tag{5}$$

Figure 3 shows the filter architecture to compute the equation 5.

In the previous section was explained how to build high order IIR-filters from a set of low order filters making use of the properties of LTI systems and interconnecting them appropriately. The same technique will be used for FIR-filters. As we know, the FIR-filters have no feedback coefficients. Due

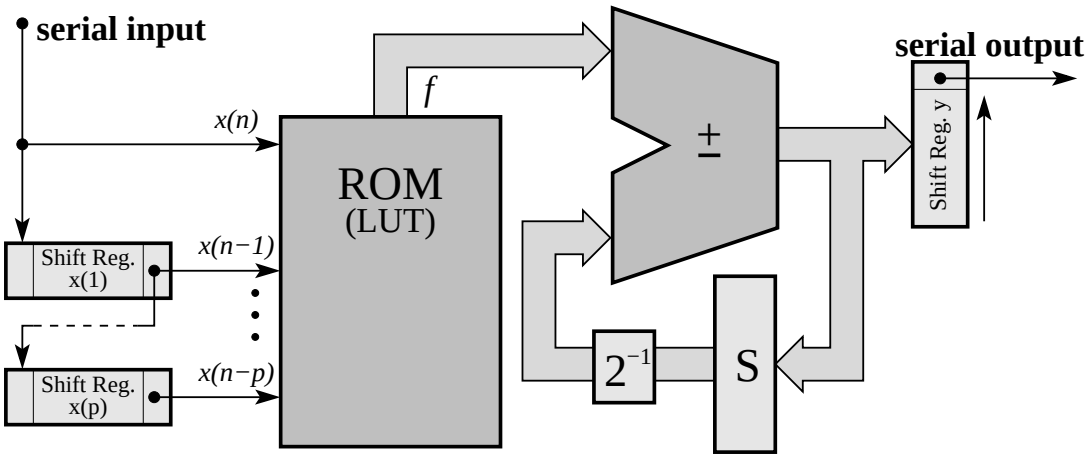


Figure 3: FIR-Filter Architecture.

to this, after $l + 1$ cycles, the *adder-substractor* output is only stored into $y(n)$ register.

To construct high order FIR-filters we need to interconnect a cascade low FIR sub-filters; in that way the input pass through them serially and the sub-filters outputs are added (by serial adders) to produce the high order FIR-filter output. The interconnection scheme is shown in Figure 4.

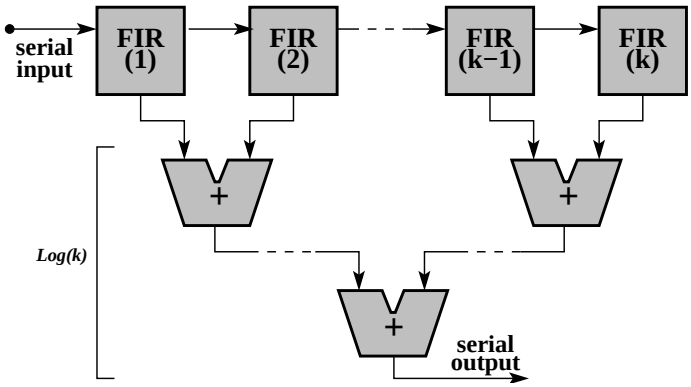


Figure 4: High Order FIR-Filter Interconnection Scheme.

If we want to construct a high order FIR-filter making use of k sub-filters, its result will have $\lceil \log_2(k) \rceil$ additional bits due to the fact that the tree adder have depth $\lceil \log_2(k) \rceil$ and each level may add one bit. Therefore, if the filter input have l bits the filter will produce one result each $l + \lceil \log_2(k) \rceil$ clock cycles.

Like it was said in the section 2, this technique reduces considerably the space required in an FPGA. For example, if we need to build a eighth-order FIR-filter we can use one fourth-order and one third-order FIR-filter. This allow us to use two small tables, a 32-entry table and a 16-entry table, instead of one 512-entry table.

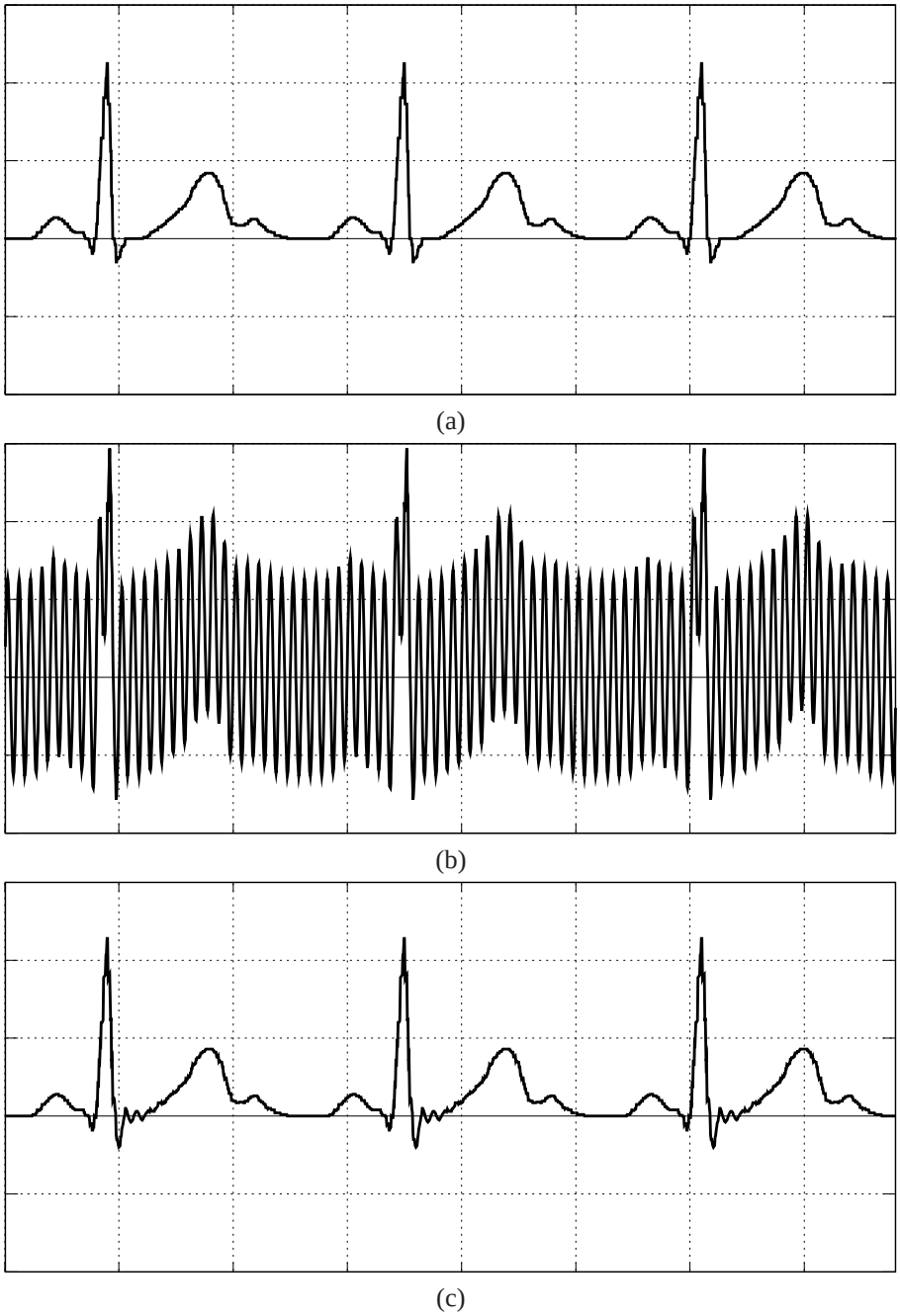


Figure 5: Removing mains-frequency interference from an electrocardiogram.

4 EXPERIMENTAL RESULTS

For experiments, several examples of filters with different order were implemented to analyze their behaviour, performance and logic cells utilization in an FPGA. All filters implemented have 8-bit input samples and their coefficients have 8-bit precision, the FPGA selected was Actel ProAsic250 series, and the ACTEL LIBERO IDE v7.3 tool was used for the synthesis. In addition, AD9102 and DAC1654 chips were used to digitize input signals and convert the output signals in analog form respectively.

Table 1: Synthesized Result.

Filter Type	Filter Order	Logic Cell	Clock Frequency
IIR	2	3.06 % (188)	112.020 MHz
IIR	4	6.28 % (386)	102.281 MHz
FIR	4	3.22 % (198)	110.156 MHz
FIR	8	6.62 % (407)	96.330 MHz
FIR	18	13.07% (803)	94.295 MHz

To see the good behaviour of the architectures presented in the previous sections we will show the functioning of a digital filter. In particular, we will consider a digital filter for an electrocardiogram. In medicine, the electrical activity of the heart can be recorded using electrodes placed on the chest, a filter can be used to reduces the fluctuations due to electric activity in the resulting electrocardiogram (60 Hz in the USA, 50 Hz in Europe). In this case the needed digital filter is a band-stop IIR-filter, because we must reject the mains supply frequency (60 Hz or 50 Hz). This filter is characterized by the following equation:

$$y^n = x^n + (-1.9021)x^{n-1} + x^{n-2} + (1.8523)y^{n-1} + (-0.94833)y^{n-2} \quad (6)$$

If the interference is at 60 Hz, the filter is effective at sampling frequency of 1200 samples per second (1.2 kHz); if it is at 50 Hz, the filter is effective at 1000 samples per second (1 kHz) [7]. The VHDL specification for this can be see in Appendix I. Figure 5 (a) shows a typical EKG waveform, corresponding to several heartbeat. In part (b) of the figure it is badly contaminated by sinusoidal interference of 60 Hz frequency. Figure 5 (c) shows the dramatic effect of this filter on the contaminated signal of part (b). The interference has been greatly reduced, without distorting the signal waveform.

Now, we will show the FPGA resources utilization of filters implemented with the techniques described in this paper. Table 1 presents these results. We can note that these techniques allow an important reduction in the logic cells utilization, also we can see that the size of filters grows lineally with the numbers of coefficients, degrading their performance slightly. We must have in mind that the overall performance of each implementation is: its clock frequency divide by the number of bits of its input signal (because this is processes serially). Therefore our implementations work at about 10 MHz, which is adequate for the most applications. These are important results, especially for FIR-filters, since they usually require many coefficients to control adequately their frequency response. In fact, using these techniques, we could synthesize a hundredth-order filter with a performance of 10 MHz approximately, it is not possible using traditional techniques with which we could synthesize sixtieth-order filters only.

5 CONCLUSION

The presented results lead to the conclusion that the use of Bit-Serial Arithmetic and Lookup Tables allow us to construct economic IIR and FIR digital filters, degrading slightly their performance. In addition, we could see that by the interconnection techniques we can construct efficient high order filters without use huge Lookup Tables. The results produced by these techniques can be straight forward translated from their schematic representation into VHDL code and then synthesize it on an FPGA. Finally, through all the examples, we could see that the behaviour of digital filters implementation is correct.

REFERENCES

[1] Rawski, Tomaszewicz, Selvaraj and Luba. “Efficient Implementation of Digital Filters with Use of Advanced Synthesis Methods Targeted FPGA Architectures”. *Digital System Design, 2005. Proceedings. 8Th Euromicro Conference on.* 30 Aug. - 3 Sept. 2005. Pages 460-466.

[2] Knut Arne Vinger and Jim Torrensen. “Implementing Evolution of FIR-Filters Efficiently in an FPGA”. *Evolvable Hardware, 2003. Proceedings. NASA/DoD Conference on.* July 9-11, 2003. Pages 26-29.

[3] Kalivas, Tsirikos, Bougas and Pekmestzi. “100% Operational Efficient Bit-Serial Programmable FIR Digital Filters”. *EUSIPCO 2005 - 13Th European Signal Processing Conference.* September 4-8, 2005. Antalya, Turkey.

[4] Chi-Jui Chou, Satish Mohanakrishnan and Joseph Evans. “FPGA Implementation of Digital Filters”. *International Conference on Signal Processing Applications and Technology.* Berlin, 1993. Pages 251-255.

[5] Sang-Hun Yoon, Jong-wha Chong and Chi-Ho Lin. “An Area Optimization Method for Digital Filter Design”. *ETRI Journal, volume 26, Number 6.* December 2004. Pages 545-553.

[6] behrooz Parhami. “Computer Arithmetic: Algorithms ans Hardware Designs”. New York: Oxford University Press, 2000.

[7] Paul a. Lynn and Wolfgang Fuerst. “Introductory Digital Signal Processing with Computer Applications”. Revised Edition. *John Wiley & Sons*. 1994.

APPENDIX I (VHDL CODE)

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity dig_filtro is port (
       x        : in  unsigned(0 to 7);
       clk      : in  std_logic;
       rst      : in  std_logic;
       y        : out unsigned(0 to 7));
end;

architecture df of dig_filtro is
  constant cBitsx      : integer := 8;
  constant cCoef       : integer := 5; -- number of coeficient
  constant cLogNumCoef : integer := 3; -- ciel of cCoef logaritm
  constant cBitsM      : integer := 8; -- number of coeficient bits
  type TableCoef_type  is array(0 to 2**cCoef-1) of
       unsigned(0 to cBitsM+cLogNumCoef-1);
  constant cTableCoef  : TableCoef_type
    :=(
       "00000000000",
       "11110000110",
       "00011101101",
       "00001110011",
       "00010000000",
       "00000000110",
       "00101101101",
       "00011110011",
       "11100001100",
       "11010010011",
       "11111111001",
       "11110000000",
       "11110001100",
       "11100010011",
       "00001111001",
       "00000000000",
       "00010000000",
       "00000000110",
       "00101101101",
       "00011110011",
       "00100000000",
       "00010000110",
       "00111101101",
       "00101110011",
       "11110001100",
       "11100010011",
       "00001111001",
       "00000000000",
       "00000001100",
       "11110010011",
       "00011111001",
       "00010000000"
);
```

```
signal x_n_reg         : unsigned(0 to cBitsx-1);
signal x_n_input       : unsigned(0 to cBitsx-1);
signal x_n_1_reg       : unsigned(0 to cBitsx-1);
signal x_n_1_input     : unsigned(0 to cBitsx-1);
signal x_n_2_reg       : unsigned(0 to cBitsx-1);
signal x_n_2_input     : unsigned(0 to cBitsx-1);
signal y_n_1_reg       : unsigned(0 to cBitsx-1);
signal y_n_1_input     : unsigned(0 to cBitsx-1);
signal y_n_2_reg       : unsigned(0 to cBitsx-1);
signal y_n_2_input     : unsigned(0 to cBitsx-1);
signal y_input         : unsigned(0 to cBitsx-1);
signal y_reg           : unsigned(0 to cbitsx-1);
signal counter_reg     : unsigned(0 to cBitsx-1);
signal counter_input   : unsigned(0 to cBitsx-1);
signal s_reg           : unsigned(0 to cBitsM+cLogNumCoef-1);
signal s_input         : unsigned(0 to cBitsM+cLogNumCoef-1);
signal f               : unsigned(0 to cBitsM+cLogNumCoef-1);
signal opndo_1         : unsigned(0 to cBitsM+cLogNumCoef-1+2);
signal opndo_2         : unsigned(0 to cBitsM+cLogNumCoef-1+2);
signal add             : unsigned(0 to cBitsM+cLogNumCoef-1+2);
signal address         : unsigned(0 to 4);


begin  -- df
  counter_input <= counter_reg(counter_reg'high) &
                   counter_reg(0 to counter_reg'high-1);

  x_n_input <= x when counter_reg(counter_reg'high)='1' else
               '0' & x_n_reg(0 to x_n_reg'high-1);
  x_n_1_input <= x_n_reg(x_n_reg'high) & x_n_1_reg(0 to x_n_reg'high-1);
  x_n_2_input <= x_n_1_reg(x_n_1_reg'high) & x_n_2_reg(0 to x_n_1_reg'high-1);
  y_n_1_input <= add(4 to 4+y_n_1_input'high) when
                 counter_reg(counter_reg'high) ='1' else
                 '0' & y_n_1_reg(0 to y_n_1_reg'high-1);
  y_n_2_input <= y_n_1_reg(y_n_1_reg'high) & y_n_2_reg(0 to y_n_1_reg'high-1);
  y_input <= add(4 to 4+y_n_1_input'high) when
             counter_reg(counter_reg'high)='1' else
             y_reg;
  y <= y_reg;

  opndo_1 <= '0' & s_reg(0) & s_reg(0 to cBitsM+cLogNumCoef-2) & '1';
  opndo_2 <= '0'&(f xor (0 to (cBitsM+cLogNumCoef-1) =>
             counter_reg(counter_reg'high)))& counter_reg(counter_reg'high);

  add <= opndo_1 + opndo_2;

  s_input <= (others => '0') when counter_reg(counter_reg'high) = '1' else
             add(1 to cBitsM+cLogNumCoef);

  address <= (x_n_reg(x_n_reg'high), x_n_1_reg(x_n_1_reg'high),
              x_n_2_reg(x_n_2_reg'high), y_n_1_reg(y_n_1_reg'high),
              y_n_2_reg(y_n_2_reg'high));

  with address select f <=
    cTableCoef(0)  when "00000",
    cTableCoef(1)  when "00001",
    cTableCoef(2)  when "00010",
```

```
    cTableCoef(3)  when "00011",
    cTableCoef(4)  when "00100",
    cTableCoef(5)  when "00101",
    cTableCoef(6)  when "00110",
    cTableCoef(7)  when "00111",
    cTableCoef(8)  when "01000",
    cTableCoef(9)  when "01001",
    cTableCoef(10) when "01010",
    cTableCoef(11) when "01011",
    cTableCoef(12) when "01100",
    cTableCoef(13) when "01101",
    cTableCoef(14) when "01110",
    cTableCoef(15) when "01111",
    cTableCoef(16) when "10000",
    cTableCoef(17) when "10001",
    cTableCoef(18) when "10010",
    cTableCoef(19) when "10011",
    cTableCoef(20) when "10100",
    cTableCoef(21) when "10101",
    cTableCoef(22) when "10110",
    cTableCoef(23) when "10111",
    cTableCoef(24) when "11000",
    cTableCoef(25) when "11001",
    cTableCoef(26) when "11010",
    cTableCoef(27) when "11011",
    cTableCoef(28) when "11100",
    cTableCoef(29) when "11101",
    cTableCoef(30) when "11110",
    cTableCoef(31) when others;

  write: process(clk,rst)
  begin
   if rst='1' then
     s_reg <= (others => '0');
     x_n_reg <= (others => '0');
     x_n_1_reg <= (others => '0');
     x_n_2_reg <= (others => '0');
     y_n_1_reg <= (others => '0');
     y_n_2_reg <= (others => '0');
     y_reg     <= (others => '0');
     counter_reg <= (0 to counter_reg'high-1 => '0',
                     counter_reg'high => '1');
   elsif clk='1' and clk'event then
     counter_reg <= counter_input;
     s_reg <= s_input;
     x_n_reg <= x_n_input;
     x_n_1_reg <= x_n_1_input;
     x_n_2_reg <= x_n_2_input;
     y_n_1_reg <= y_n_1_input;
     y_n_2_reg <= y_n_2_input;
     y_reg <= y_input;
   end if;
  end process;
end df;
```

Implementación de DCCP (Datagram Congestion Control Protocol) en espacio usuario

Guillermo Rigotti
UNICEN – Fac. de Ciencias Exactas-ISISTAN
Pje. Arroyo Seco, (7000) Tandil, Bs. As. Argentina
TE: +54-2293-439682 FAX: +54-2293-439681 Email: grigotti@exa.unicen.edu.ar

Abstract

Until recently time, TCP and UDP satisfied application´s requirements. This situation changed when real time multimedia applications appeared. As a consequence of its reliability, TCP does not enable applications to control the sending rate. This produces delays that are incompatible with the service to be offered by the application. UDP enables applications to control the sending rate; this is dangerous for the network, because it could result congested. To solve this situation, the IETF defined DCCP*(Datagram Congestion Control Protocol)*, a protocol specifically designed to support real time multimedia applications.
DCCP offers basic functionality: connection establishment and termination, support to incorporate congestion control strategies, and options negotiation. This allows it to adapt efficiently to the requirements of the different types of multimedia real time applications. Currently, a significant research activity related to DCCP is being carried out in the IETF. Among others, there are two main topics that deserve special attention, and that are being object of research: the definition of an application API and the incorporation and testing of congestion control mechanisms that can adapt to various types of applications. This paper presents an implementation of DCCP coded in Java, running in user space. Its main characteristics are portability and modularity. The goal of our work is to provide a way to experiment and to test the two aforementioned topics, application API and congestion control mechanisms. This alternative allows work with DCCP and the applications in any environment and without any risk for the operating system.
Keywords: DCCP, congestion control, real time multimedia applications.

Resumen

Hasta hace poco tiempo, el soporte de comunicaciones de TCP/IP, TCP y UDP, fue suficiente para satisfacer los requerimientos de las aplicaciones. Esto cambió con la aparición de las aplicaciones multimedia en tiempo real. TCP no permite a la aplicación controlar la tasa de envío, ocasionando demoras incompatibles con el servicio ofrecido. UDP permite controlar la tasa de envío, pero resulta peligroso ya que no provee control de congestión, pudiendo saturar de la red. Como respuesta a esta situación, la IETF definió DCCP (Datagram Congestion Control Protocol), un protocolo diseñado para soportar aplicaciones multimedia de tiempo real. DCCP se caracteriza por ofrecer mínima funcionalidad: conexión, soporte para control de congestión, y negociación de opciones de operación. Esto le permite adaptarse eficientemente a los requerimientos de los diferentes tipos de aplicaciones multimedia en tiempo real. Actualmente existe una actividad de investigación importante referida a DCCP en el ámbito de la IETF. Entre otros, hay dos aspectos relacionados con DCCP que merecen atención y que son objeto de investigación: la definición de una API a la aplicación y la incorporación y prueba de nuevos mecanismos de control de congestión. En este paper se presenta una implementación en Java de DCCP, en espacio usuario, cuyas principales características son la portabilidad y modularidad. El objetivo es posibilitar la rápida experimentación y prueba en los dos aspectos mencionados, API y métodos de control de congestión. Esta alternativa permite trabajar con DCCP y las aplicaciones en cualquier medioambiente y sin riesgos para el sistema operativo.
Palabras clave: DCCP, control de congestión, aplicaciones multimedia de tiempo real.

1. Introducción

DCCP (Datagram Congestion Control Protocol) [1] es un protocolo de nivel transporte que se originó en el ámbito de la Internet Engeneering Task Force (IETF) [2] como respuesta a la aparición de tipos de aplicaciones no tradicionales[1] que requieren servicios de transporte de datos diferentes a los ofrecidos por TCP[3] y UDP[4].

La actividad de investigación referida a DCCP se lleva a cabo principalmente en el DCCP Working Group, desde 2001. Actualmente se cuenta con un conjunto de standards (RFCs) que cubren diferentes aspectos del protocolo: especificación, alternativas de control de congestión, especificaciones de uso de DCCP con aplicaciones existentes y nuevas, y agregado de funcionalidad modular al protocolo, que lo haga atractivo a dichas aplicaciones.

El tipo de aplicaciones para el que fue diseñado DCCP, se caracteriza por generar flujos de información de larga duración, ser sensible a las demoras y tolerar pérdida o corrupción en los datos. El soporte tradicional ofrecido a nivel transporte, materializado por TCP y UDP no satisface estos requerimientos.

TCP se caracteriza por ser un protocolo que provee un servicio orientado a conexión, modo stream y confiable. Si bien el establecimiento de conexión es necesario para controlar el flujo de datos originado por las aplicaciones para las que fue concebido DCCP, la característica de confiabilidad y entrega ordenada de la información provista por TCP provoca efectos no deseados: por ejemplo, en una transmisión telefónica, es preferible recibir un datagram con datos erróneos, lo que provocará una interferencia no significativa para el receptor, que esperar el tiempo de recuperación impuesto por TCP, lo cual impediría la comunicación en tiempo real. Por otro lado, los mecanismos internos utilizados por TCP no son fácilmente adaptables a los requerimientos de DCCP, ya que los aspectos de confiabilidad y control de congestión se encuentran altamente relacionados.

UDP se caracteriza por ofrecer un servicio no orientado a conexión, no confiable. Las ventajas que presenta son su bajo overhead y la no interferencia con el envío de datos realizado por la aplicación, no imponiendo restricciones en la tasa de envío. Desde sus comienzos, UDP ha sido utilizado por aplicaciones que ocasionan poco tráfico en la red. Actualmente, es la única opción para soportar las aplicaciones multimedia de tiempo real, debido a las características no deseables de TCP. El problema que se presenta es que al no controlarse a nivel transporte el tráfico que se inyecta en la red, este tipo de aplicaciones pueden causar congestión, en la medida en que constituyan una parte significativa del tráfico en la Internet.

La alternativa de implementar control de congestión a nivel aplicación, presenta dos problemas, uno de ellos es que no siempre las aplicaciones lo implementarán, ya que es un aspecto complicado y sujeto a errores; por otra parte, en caso de que lo hicieran, seria un replica de la funcionalidad innecesaria.

Como respuesta a esta situación, surge DCCP, que es un protocolo orientado a conexión, de muy bajo overhead y adaptable a los diferentes requerimientos de las aplicaciones a través de mecanismos de negociación de opciones de operación. Entre las opciones, puede negociarse el tipo de control de congestión a utilizar. Si bien aún no esta ampliamente difundido, la expectativa es que sea el protocolo que reemplace a UDP para aplicaciones multimedia de tiempo real.

En la actualidad, DCCP se encuentra implementado en el kernel Linux 2.6.14 y posteriores [5] y en estado de desarrollo en BSD [6]. Existe además una implementación en lenguaje C, en espacio usuario en Linux [7], cuyo objetivo es ganar experiencia en varios aspectos del protocolo a través de un desarrollo rápido para luego portar la implementación al kernel.

En este trabajo se presenta una implementación de DCCP en espacio usuario, realizada en lenguaje Java, lo cual la hace portable a cualquier plataforma.

[1] Telefonía IP, multiconferencia, y en general aplicaciones multimedia de tiempo real.

Uno de los objetivos perseguidos consiste en proveer una base para experimentar con aplicaciones que por sus requerimientos sean candidatos a utilizar DCCP., en aspectos tales como la definición de APIs a nivel usuario (DCCP aplicación [8] [9]).
Otro aspecto de importancia lo constituye el desarrollo de heurísticas de control de congestión y su evaluación, en cuanto a su adaptabilidad a las aplicaciones y su compatibilidad con tráfico TCP.
Lo que distingue a esta implementación es su portabilidad, lo cual permite el uso y experimentación con DCCP entre cualquier par de equipos conectados a la Internet. Este aspecto es de importancia dada la escasa difusión que aún caracteriza a este protocolo..
El resto del trabajo está organizado de la siguiente manera: en la sección 2 se describen brevemente las características principales de DCCP, resaltando sus aspectos más importantes; en la sección 3 se describe la implementación presentada en este trabajo, de manera general; posteriormente, en las secciones 4, 5 y 6 se describen aspectos más específicos de la misma, respectivamente: threads de ejecución, interacción con las aplicaciones, y los módulos principales que componen la implementación. La sección 7 se refiere a las conclusiones, estado actual y continuación del trabajo. La sección 8 contiene las referencias bibliográficas.

2. Descripción de DCCP

DCCP es un protocolo de nivel transporte, al igual que TCP y UDP. Provee un servicio de transmisión de paquetes orientado a la conexión, no confiable y con control de congestión. Este protocolo provee la funcionalidad mínima necesaria, con el objeto de reducir al máximo el overhead y mantener un diseño simple. Deja a las aplicaciones funciones tales como FEC (forward error correction) y manejo de múltiples streams entre otras.
En lo que sigue de esta breve descripción, se remarcarán las diferencias con TCP; de esta manera se trata de aclarar la razón por la cual este protocolo no puede ser un subconjunto ni un derivado de aquél.

2.1 Formato de paquetes

DCCP define varios tipos de paquete, cada uno de ellos con funciones específicas, evitando el uso de los bits de flag, como hace TCP en su único formato de segmento.
Los paquetes DCCP tienen un formato simple, para su rápido proceso, permitiendo el agregado de opciones para diferentes funciones. Las opciones pueden extenderse hasta 1008 bytes. En este aspecto es más flexible que TCP, que tiene un campo de opciones limitado en longitud. En la figura 1 se muestra un paquete DCCP. Por razones de extensión, sólo se muestra el caso de números de secuencia extendidos, de 48 bits, en el caso de números de secuencia de 24 bits, se mantienen los campos mostrados. Los *ports* de origen y destino, al igual que en TCP, son los que juntamente con las direcciones IP determinan los endpoints de la conexión. El campo *data offset* indica dónde comienza, si hay, la zona de datos de la aplicación, determinando también la longitud del campo de opciones. *CCVal* es un valor de uso opcional, destinado al método de control de congestión, por lo que no se define aquí su función. *CsCov* indica qué parte del segmento DCCP está cubierta por el campo *checksum*:.además del header fijo y las opciones Puede especificarse qué parte de los datos de la aplicación deben ser cubiertos por el checksum, dependiendo de la sensibilidad de la misma a datos corruptos. El campo *Type* determina el tipo de paquete DCCP: hasta el momento hay definidos 10, los cuales se muestran en la figura 1. El campo *X* especifica si se usan números de secuencia cortos o largos, y por último, el campo *secuencia* especifica el número de secuencia del segmento DCCP.
En los segmentos DCCP_Request y DCCP_Response, se define un campo *Service Code*, que hace referencia a la aplicación a la cual sirve DCCP en el port (por ejemplo, RTP).
Todos los segmentos excepto DCCP_Request y DCCP_Data, llevan un subheader que contiene un número de Ack, que hace referencia al número de secuencia del paquete con mayor número y con header correcto que se ha recibido.

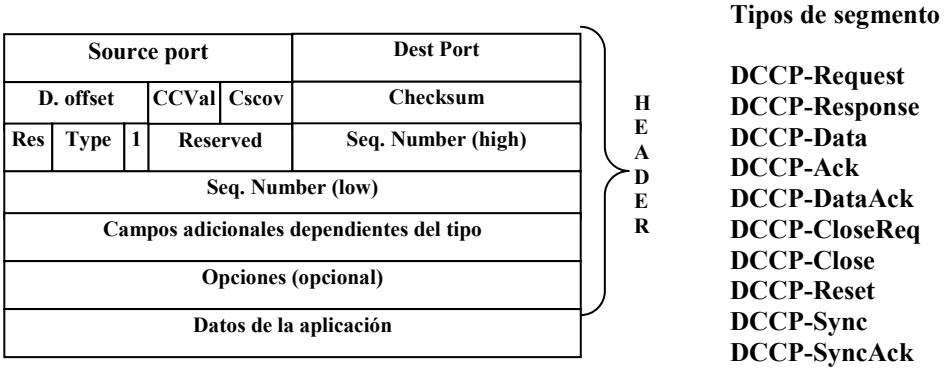


Fig. 1 Formato de segmentos DCCP

2,2 Funciones de los segmentos

Los segmentos que se utilizan para el establecimiento de la conexión son DCCP_Request, DCCP_Response y DCCP_Ack. Durante esta etapa se lleva a cabo normalmente la negociación de opciones de operación.

La terminación de la conexión se realiza, de manera normal, utilizando los segmentos DCCP_Close, DCCP_CloseReq (sólo el servidor) y DCCP_Reset. Esta último determina además el cierre incondicional de una conexión, debido a un error, por ejemplo la imposibilidad de negociar una opción (mandatory).

Los segmentos DCCP_Data, DCCP_DataAck .y DCCP_Ack se utilizan para el intercambio de datos, datos y asentimientos y asentimientos respectivamente.

Los segmentos DCCP_Sync y DCCP_SyncAck se utilizan para recuperar al DCCP de una situación de error, como conexiones parcialmente establecidas o grandes ráfagas de segmentos perdidos.

2.3 Manejo de conexiones

La conexión se establece a través del intercambio de paquetes específicos, (DCCP_Request, DCCP_Response y DCCP_Ack), no soportando conexión simultánea.

La terminación de la conexión involucra el intercambio de segmentos DCCP_CloseReq, DCCP_Close y DCCP_Reset. Ambos procedimientos son confiables.

Una conexión bidireccional es considerada separada en dos conexiones unidireccionales, comprendiendo cada una de ellas los datos que viajan en un sentido, y los asentimientos correspondientes a ellos, que viajan en el otro. Los datos y asentimientos de ambas conexiones unidireccionales se mezclan en los segmentos DCCP (piggybacking). La particularidad de considerar dos conexiones separadas es que cada una de ellas funciona independientemente de la otra en cuanto a los parámetros de operación y al método de control de congestión utilizado. Esta característica es de suna importancia en el tipo de aplicaciones al que está orientado DCCP. La opción *Init Cookie* permite al servidor no guardar estado de una conexión en vías de establecimiento.

2.4 Opciones

El manejo de opciones provisto por DCCP tiene una considerable importancia, ya que permite parametrizar el funcionamiento del protocolo, seleccionar el método de control de congestión a utilizar, y chequear el estado de la red, entre otras cosas. De esta manera, DCCP puede adaptarse a los diferentes tipos de aplicaciones que lo utilizan.

Desde el punto de vista del manejo de las opciones por parte del protocolo, podemos distinguir aquellas opciones que son procesadas por el receptor (por ejemplo el uso de checksum), aquellas que además implican una respuesta (a través del envío de otra opción – por ejemplo timestamp y timestamp echo-) por parte de este último, y aquellas que se refieres a negociación de parámetros de funcionamiento del protocolo (features).

Todos los intercambios de opciones pueden realizarse en cualquier fase del protocolo, pero los de negociación, generalmente se realizan en la fase de conexión..
Las características negociables del protocolo, denominadas features, se representan como valores almacenados y posibles de ser negociados con la otra parte y/o modificados a requerimiento de la aplicación. Por ejemplo, una feature se refiere al mecanismo de control de congestión a utilizar, teniendo almacenados dos valores, uno que representa el preferido, y otro alternativo. El mecanismo de negociación de opciones es simple y adaptado a las características de no confiabilidad del protocolo. Se basa en el intercambio de dos tipos de opción, Change, con la cual un lado indica al otro que desea cambiar el valor de una feature determinada y Confirm, que indica una respuesta (positiva o no) al Change. Las features son almacenadas en cada uno de los lados, con preferencias que pueden diferir. Las opciones Change y Confirm, deben referirse a una feature local al que inicia la solicitud de cambio, o remota, para lo cual se utilizan respectivamente los pares (Change_L, Confirm_R y Change_R, Confirm_L).
La confiabilidad de la negociación de opciones se basa en la repetición (limitada) de cada Change hasta obtener el correspondiente Confirm, y una protección contra opciones reordenadas en la red, basada en los números de secuencia de los segmentos.

2.5 Control de congestión

Como consecuencia de haber sido diseñado para soportar una diversidad de aplicaciones con diferentes requerimientos de control de congestión, DCCP soporta la negociación del método de control de congestión a ser utilizado, destinando además un porcentaje significativo (aproximadamente un 50%) de códigos de falla (reset), features y opciones para su uso por parte de métodos específicos de control de congestión. El método de control de congestión a utilizar se selecciona independientemente para cada conexión unidireccional de acuerdo a las necesidades de la aplicación.
Actualmente hay dos definidos : el identificado como CCID 2 [10] ofrece un control de congestión similar al de TCP, utilizando una ventana de congestión para mantener los paquetes enviados sin asentimiento. Cuando detecta congestión (paquetes descartados o marcados ECN) baja a la mitad la tasa de envío. Para determinar qué segmentos fueron bien recibidos en una ventana, se utiliza un mecanismo similar a SACKs.[11].
CCID 3 [12] aplica TCP-Friendly Rate Control (TFRC). El emisor mantiene una tasa de transmisión que va actualizando con las estimaciones del receptor acerca de paquetes erróneos y marcados como ECN.
Debe tenerse en cuenta que si bien ambos métodos de control de congestión son "TCP friendly"[2], esto no es un requisito impuesto por DCCP,

3. Generalidades de la implementación de DCCP

El trabajo presentado consiste en la implementación de la funcionalidad de DCCP de acuerdo al standard [1]. Esta implementación, realizada en Java, tiene las características de ser portable y modular.
La portabilidad fue un aspecto al que se dio fundamental importancia, ya que de esta manera, el protocolo puede ser ejecutado en cualquier plataforma que soporte Java, pudiendo por lo tanto probarse entre cualquier par de equipos conectados a la Internet, y no sólo aquellos con sistemas operativos que incluyan soporte DCCP (Linux con kernel 2.6 o BSD –experimental-)
La modularidad permite incluir fácilmente nuevos métodos de control de congestión y opciones[3], sin tener que modificar el código del DCCP.

[2] "TCP friendly" se llama a un protocolo que en el largo plazo no excede el ancho de banda normal usado por una conexión TCP en igules condiciones.
[3] La inclusión de este tipo de funciones es un aspecto importante del protocolo.

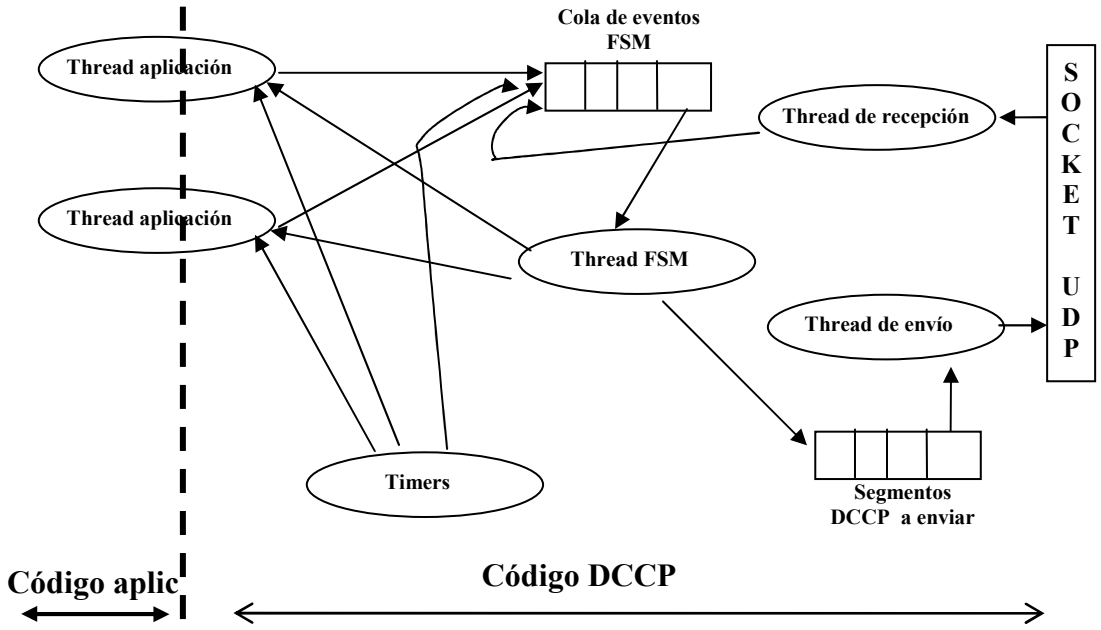


Fig. 2. Esquema simplificado de los threads que componen la totalidad del proceso (DCCP más aplicación). Los threads se muestran como elipses y sus interacciones con flechas. Las interacciones son sincronizadas. Los threads de la aplicación (uno o más) se bloquean en el código DCCP hasta que son satisfechos sus requerimientos o son informados de otras situaciones a través de excepciones

Al estar implementado en espacio usuario, las modificaciones realizadas no representan riesgos para el sistema, y pueden realizarse de manera rápida y sencilla. Posteriormente, dicha funcionalidad podrá ser portada al kernel[4].

Una vez completo, la implementación de DCCP permitirá investigar sobre aspectos de importancia del protocolo, casi sin abordar en la actualidad, como los siguientes:

Desarrollo de nuevos métodos de control de congestión orientados a necesidades de diferentes tipos de aplicaciones.

Evaluación de dichos métodos en cuanto a su adaptación a la aplicación, y en cuanto al uso que hacen de la red y de qué manera son compatibles con el tráfico TCP.

Desarrollo de una API a nivel usuario (DCCP/Aplicación). Este aspecto debe ser aún analizado ya que los requerimientos de las aplicaciones difieren significativamente, por ejemplo algunas necesitarán saber el estado en que llegaros ciertos bloques o si algunos fueron marcados como ECN, etc.

Experimentación y definición de la manera de utilizar DCCP por parte de otros protocolos, por ejemplo Real Time Protocol (RTP) [13]..

Se tomó la decisión de utilizar UDP como soporte de comunicaciones, debido a que este protocolo no impone restricciones de envío ni genera un overhead significativo. La otra alternativa, uso de IP en forma directa (raw sockets) fue considerada, pero se descartó ya que no es posible disponer de librerías Java standard para el acceso al nivel de red. Estas no son provistas por Java debido a que para algunos, resultaría en un problema de seguridad, mientras que otros argumentan que seria muy útil ya que pondría a Java a la altura de otros lenguajes como C++ en aspectos relativos a aplicaciones de monitoreo de red y similares. Un problema que surge de la utilización de UDP, es que a nivel sockets no se puede proveer una indicación de congestión (ECN), sin embargo, no todos los routers implementan dicha característica.

[4] En los casos en que se cuente con soporte DCCP en el sistema operativo.

Otra limitación que surge del uso de UDP es la imposibilidad de comunicación con versiones de DCCP standard Sin embargo, previendo la necesidad de esta prueba, se utilizará algún soporte de amplia difusión para acceso al nivel de red, tal como libpcap [14] desde un ambiente Linux, con una librería de acceso desde Java (Jpcap [15]), a efectos de poder interactuar directamente con IP y confrontar este desarrollo con implementaciones standard de DCCP.
A pesar de encapsular el segmento DCCP en UDP, se mantienen exactamente los formatos de paquetes en cuanto a la longitud de los campos, para no introducir mayor overhead y poder utilizar longitudes razonables para los segmentos DCCP,

4. Threads de ejecución

Para lograr una implementación clara y modular, se utilizó la capacidad de threads provista por Java. Se definió de esta manera un thread de lectura y otro de escritura sobre el socket UDP utilizado para comunicación, encargados respectivamente de recibir/desencapsular y encapsular/enviar los segmentos DCCP contenidos en los datagrams UDP.
Un thread adicional, implementa la FSM definida en [1], con sus 9 estados; este thread procesa eventos depositados por los demás threads en una cola de eventos en forma continua, dando lugar a los procesos DCCP (incluidas las invocaciones a métodos de control de congestión, negociación de opciones, etc).
Por último, uno o más threads definidos por la aplicación, ejecutan código DCCP (encargado de atender requerimientos de la aplicación) desencadenando eventos que son colocados en la cola de la FSM para su proceso. En ls figura 2 se muestra la interacción entre los threads mencionados.
Además de los threads mencionados, debemos considerar el proceso asincrónico realizado por los timers, ya que estos pueden afectar variables utilizadas por los primeros.

5. Interacción DCCP/Aplicación

En el caso standard, un DCCP implementado en espacio kernel, la interacción entre éste y la aplicación se realiza a través de sockets, como en TCP y UDP. Sin embargo, a diferencia de lo que ocurre en estos últimos, en el caso de DCCP dicha interacción se vuelve más compleja debido a los requerimientos especiales de cada tipo de aplicación.
Por ejemplo, en una aplicación que tolera la pérdida de paquetes, se podría requerir de DCCP una información detallada acerca de qué paquetes se han perdido, o bien la aplicación podría querer informar a DCCP qué paquetes ha descartado debido a su llegada fuera de término,
Otro aspecto de esta interacción, es el relacionado con la configuración inicial (incluido el método de control de congestión a utilizar) y le especificación de modos de operación por parte de la aplicación, que da lugar a la negociación de opciones durante el curso de la conexión.
Estos aspectos de la interacción no están aún definidos, debido a su complejidad y a la variedad de aplicaciones y requerimientos. Algunos intentos de definir esta API, que implica la modificación de los llamados a los sockets, pueden encontrarse en [16]
En el caso de la implementación que se presenta, estos aspectos se encuentran aún casi sin definir, habiendo implementado los mecanismos básicos que permiten una interacción simple de DCCP con las aplicaciones.
Para este mecanismo de interacción entre DCCP y la aplicación; se optó por proveer un conjunto de llamadas bloqueantes, con la misma filosofía de java, y el uso de excepciones (DCCPException) para informar al thread de la aplicación de situaciones anormales. En el caso de contar con un soporte DCCP en el kernel y librerías Java que soportaran un API apropiada sobre los sockets, la migración de las aplicaciones sería casi inmediata.
La funcionalidad definida hasta el momento, es la siguiente:

- Especificación de el o los Sevice Codes, a efectos de poder especificar el o los tipos de aplicación soportados por el port[5].

[5] Uno en el caso de un cliente, y uno o más en el caso de un servidor.

- Indicación a la aplicación de la señal de receptor lento (Slow Receiver) para que la aplicación emisora pueda aplicar un comportamiento acorde.
- Especificación por parte de la aplicación, de un número máximo de reintentos de negociación de opciones, para no crear ciclos.
- Especificación por parte de la aplicación de la semántica de descarte de paquetes
- Especificación por parte de la aplicación, del cambio de lista de preferencias para una opción negociable determinada.

6. Módulos que componen la implementación

En esta sección se describen los módulos de mayor importancia. Algunos de ellos se corresponden con threads descriptos anteriormente (por ejemplo la máquina de estados DCCP, y los módulos de emisión y recepción**).**

6.1 Inicialización e interfaz con la aplicación

Las funciones de inicialización se realizan al crear la instancia DCCP. Consisten en la creación de los threads, en la inicialización de variables del generales del DCCP y en la configuración del modo de operación por defecto; éste podrá ser posteriormente modificado por la aplicación.

La función más importante de este módulo es servir de nexo entre la aplicación y el DCCP, y sincronizar su operación. Las invocaciones de la aplicación recibidas por este módulo son tratadas de manera tal que sean compatibles con las invocaciones provistas por java al nivel de transporte, es decir, llamadas bloqueantes con anuncio de situaciones anormales a través del mecanismo de excepciones, en este caso, utilizando el tipo DCCPException.

En estas interacciones deben considerarse dos aspectos: el mecanismo de la interacción, que se encuentra totalmente definido, y las interacciones en sí, que irán siendo definidas a medida que se defina un API entre el DCCP y la aplicación; este aspecto es objeto de investigación [16].

La sincronización entre el thread de la aplicación y la FSM DCCP se realiza de la siguiente manera: el thread de la aplicación ejecuta código de la aplicación hasta que produce una invocación a DCCP (por ejemplo read, connect o cambio en la lista de preferencias para una feature); en este momento, se produce un evento en la FSM DCCP, de acuerdo a lo solicitado por la aplicación: por ejemplo la invocación a read produce un evento receive a ser procesado por la FSM. Una vez colocado el evento en la cola de eventos de la FSM, el thread originado por la aplicación quedará esperando la resolución de lo solicitado a DCCP. Dicha espera se produce a través del testeo de una variable incluida en el evento producido, pudiendo finalizar por éxito en lo solicitado o a través de una excepción, si se producen condiciones anormales[6]. Dicha excepción deberá ser tratada por código provisto por la aplicación.

Hay casos en los que la llamada a DCCP se resuelve inmediatamente, por ejemplo cuando se requiere información acerca del valor de una variable local[7]. En otros casos, el evento producido puede o no dar una respuesta a la aplicación. Por ejemplo, en el caso de un read, es posible que cuando se genera el evento correspondiente, la FSM tenga un frame de la aplicación listo para ser leído, que se devuelve a la aplicación, con lo cual finaliza la invocación con el fin del proceso del evento. Sin embargo, podría ocurrir que aún no hubiera un frame disponible, y como el llamado es bloqueante, se debería esperar hasta el arribo o hasta que venza el timer de espera de la aplicación. En ninguno de estos casos es posible demorar el evento de la FSM hasta que se produzca alguna de las alternativas. La solución que se adopta consiste en que el código invocado por la aplicación quede en un ciclo, en el cual realiza repetidas invocaciones a la FSM, a intervalos regulares, hasta que se tiene éxito o se produce una condición de error. De esta manera, los eventos se resuelven a ritmo de la FSM, sin interferir con otros que ella deba procesar. Los posibles casos en un read son la

[6] Se mantiene en todos los casos el mecanismo usual de retorno de llamadas a las funciones de comunicación utilizado por Java.

[7] En estos casos, en general, no es necesario producir un evento a la FSM, resolviéndose la invocación a través de una consulta a una variable.

recepción de los datos, retorno normal, o un error de recepción tardía, que sería indicado por la correspondiente DCCPException..
Otro caso lo constituye el de un open, ya que aquí hay un único evento producido como consecuencia de la aplicación, pero debe esperarse por una reacción del servidor DCCP. No es posible ni tiene sentido generar repetidos eventos open, sino que se debe memorizar el evento original, para luego poder entregar el resultado. Para lograr esto, se introduce una variable en el código de la FSM, que recuerda el evento open; luego, cuando se recibe el evento correspondiente, que determina si el open fue o no aceptado, la FSM lo anunciará al código de invocación de DCCP, quien procederá de la misma manera que la descripta en el caso anterior. Debe notarse que es posible tener múltiples invocaciones concurrentes en el caso de un read o write, pero se permite una única invocación a open[8].

6.2 Máquina de estados DCCP

Este módulo implementa la máquina de estados definida en [4340], compuesta de 9 estados que incluyen las fases de establecimiento y terminación de la conexión y la transferencia de datos. Se agregan estados y eventos propios de la implementación por razones de eficiencia.
Este código es ejecutado continuamente, procesando eventos almacenados en una cola a ese efecto; estos eventos corresponden a los siguientes sucesos:
Llegada de segmentos DCCP remotos, entregados por el thread de recepción. Estos son analizados y de acuerdo al estado de la FSM y al tipo de segmento, producen cambios y/o son entregados a otros módulos para continuar su proceso (por ejemplo al módulo de manejo de opciones o al de control de congestión, según contengan opciones o datos y/o asentimientos).
Vencimiento de timers directamente relacionados con la operación de la FSM (no incluye otros tales como timers de retransmisión para negociación de opciones, etc.). Estos eventos podrán producir, dependiendo del caso, cambio de estado de la FSM, notificaciones a la aplicación, o cierre de la conexión.
Eventos producidos por el módulo de interfaz con la aplicación como consecuencia de requerimientos de esta última. En algunos casos, estos eventos producirán cambios de estado en la FSM.
Además de los mencionados, se agregaron dos eventos propios de la implementación para mejorar la eficiencia y simplificar el código. Se utilizan en situaciones especiales, que como en todos los casos requieren un proceso ordenado de los eventos ya producidos y aún no ejecutados por la FSM. Estos eventos, que no alteran el comportamiento descrito en [1], son CANCEL_INCOMING , que es producido cuando desde la aplicación se cancela un Listen previo (sólo en el servidor), con lo cual se debe volver al estado inicial, y RESET que indica una condición excepcional que hará que la FSM prepare la terminación de la conexión, enviando la causa del reset a todos los eventos pendientes.
Cada evento consta de un código que lo identifica, un resultado de la operación que es colocado por la FSM, y un objeto dependiente del tipo de evento. En algunos casos de eventos generados por la aplicación, éstos conservan el resultado de la operación y son consultados por aquella luego de su ejecución por parte de la FSM (por ejemplo, establecimiento de la conexión).
En resumen, los eventos definidos hasta el momento, son los siguientes.
1-Los producidos por el thread de al aplicación, se corresponden con las llamadas que ésta hace a DCCP: por el momento. Algunos de estos eventos están relacionados con la API DCCP/aplicación.
2-Evento de recepción de segmentos, generado por el thread de recepción, que contiene el segmento DCCP recibido desde el lado remoto.
3-Los eventos internos de la implementación para proceso ordenado de los eventos ante un cierre de la conexión o cancelación de espera por requerimientos de conexión (RESET y CANCEL_INCOMING).

[8] Esto es controlado por la FSM, al analizar el par estado/evento.

4-El evento generado por el timer que controla, en ciertos casos de terminación de conexión, el tiempo de espera para la reutilización del port involucrado (TIMEWAIT).

6.3. Manejo de opciones

Como fue mencionado en la sección 2.4, una de las características de DCCP es su versatilidad en el manejo de opciones, lo que le permite, entre otras de sus características, adaptarse a diferentes tipos de aplicación. Una descripción detallada de las opciones definidas hasta el momento se encuentra en [1][9].

El manejo de opciones presenta características que justifican su implementación por separado del resto de los módulos:

- Opciones que requieren proceso dentro del módulo
- Opciones de negociación de features, que requieren timers y reintentos.
- Envío de opciones sujeto al control de flujo en uso pero independiente de los datos y/o segmentos a enviar.
- Interacción con la aplicación para la selección de valores de features.
- Previsión de agregado de nuevas opciones y features.
- Diferentes procesos de negociación de features.

A continuación se describen las interacciones entre el manejo de opciones y el resto de la implementación.
En el arranque de DCCP, se invocan los métodos de inicialización de opciones y de features; éstos se encargan de inicializar las opciones implementadas, con los valores por defecto. Posteriormente, se realizará la posible negociación de algunos valores (features); esta negociación se dará al comienzo debido a requerimientos de la aplicación (especificadas al solicitar la conexión a travès de la API a definir) o a diferencias de valores iniciales entre ambos DCCPs, o durante el transcurso de la conexión, debido a requerimientos de las aplicaciones o bien a la recepción de opciones de negociación del lado remoto. .
El uso de otras opciones (por ejemplo las de medición de tiempos), estará determinado por el propio DCCP, y por el método de control de congestión en uso.
Por el momento y debido a la funcionalidad de las opciones definidas en [1] se decidió implementarlas en el módulo de opciones y no por separado. Un caso especial es el método de control de congestión utilizado, que se implementa de manera independiente.
La comunicación entre el módulo de manejo de opciones y el envío de los segmentos, se produce a través de una lista de opciones a enviar.
Para el envío de opciones, la función de construcción de segmentos DCCP (invocada por la FSM y sujeta al control de congestión utilizado) consulta la lista de opciones a enviar, agregándolas en el segmento a enviar, siempre que sean aceptables para el tipo de segmento.
Las opciones, son generadas a través de requerimientos al módulo de opciones. Estos requerimientos son realizados por el módulo de negociación, por requerimientos del DCCP (por ejemplo si se ha configurado el uso de NDP), por requerimientos de la aplicación, que pueden dar lugar al cambio de la lista de preferencias para alguna opción negociable, o por el método de control de congestión en uso (opciones reservadas para el CCID).
A continuación se describen de manera simplificada las interacciones entre el módulo de manejo de opciones y el resto de la implementación

1. Llegada segmento DCCP (**recepción)**
   1.1 separación de opciones (**FSM DCCP**)

[9] Se prevé que al ir extendiéndose DCCP a soportar diferentes tipos de aplicación, se cree una cantidad considerable de nuevas opciones y features; esto se encuentra reflejado en el significativo porcentaje de valores reservados para los dos tipos.

1.2 proceso de opciones remotas (**opciones**)
1.2.1 Modificación de parámetros de funcionamiento (p.ej cambio de NDP)
1.2.2 Generación de opción de respuesta (p. ej. timestamp echo)(**opciones**)
1.2.3 Generación de opciones de respuesta a una negociación (Confirms) (**opciones**)
2. Vencimiento de timer de envío de Change
2.1 Generación de nuevo Change, si no esta en lista de opciones a enviar(**opciones**)
3. Modificación de lista de preferencias (aplicación u otro módulo)
3.1 Invocación a la feature, y luego posible generación de opción Change (**opciones)**
4. Envío de segmento (FSM). Extracción de opciones de la lista **(opciones)**

6.4. Emisión y recepción

Estos módulos se corresponden con los respectivos threads, y su función es la comunicación del DCCP local con el remoto.
El módulo de emisión inicializa el socket UDP de envío, y luego se ejecuta continuamente generando segmentos DCCP en función de datos e información de control suministrados ya sea por la FSM o por el método de control de congestión, y colocados en una cola de segmentos a enviar. De acuerdo las características del control de congestión, el envío puede ser bloqueado temporalmente.
El thread de recepción, sólo se encarga de desencapsular información recibida por el socket UDP, y entregarla, en forma de evento, a la FSM. Sería posible incluir en él un chequeo que permita descartar segmentos no válidos (independientemente de las condiciones del DCCP local) que de todas maneras serán ignorados, para mejorar la eficiencia.

6.5. Control de congestión

El módulo de control de congestión no se trata aquí debido a que depende del tipo de aplicación soportado. De acuerdo a sus necesidades, se selecciona el adecuado durante el establecimiento de la conexión. Los dos tipos de control de congestión definidos por el grupo de trabajo de DCCP [10] y [12] están actualmente siendo implementados.

7. Estado actual, conclusiones y continuación

Actualmente, se encuentran implementados y en funcionamiento los mecanismos de interacción DCCP/Aplicación, el establecimiento y la terminación de conexión, el manejo de opciones (incluido el código de las más simples) y el envío de segmentos DCCP. Se encuentran en estado de desarrollo los métodos de control de congestión ya mencionados, previéndose que cuando se los incorpore al protocolo se ajustará la interacción entre éste y los primeros.
La aplicación desarrollada será útil para experimentar con el API ofrecido a las aplicaciones y para probar las heurísticas existentes de control de congestión, y posiblemente desarrollar otras para nuevos requerimientos de las aplicaciones. Estos aspectos son de importancia ya que DCCP no está aún muy difundido.
Un problema de la implementación consiste en que DCCP se encapsula en UDP, no pudiendo de esta manera comunicarse con implementaciones DCCP en el kernel, que se encapsulan en IP. Otra consecuencia del uso de UDP es que no es posible acceder a información de IP, en particular a los bits ECN, con lo cual algunos métodos de control de congestión perderían información útil para tomar sus decisiones. Para salvar estos inconvenientes, sería posible utilizar librerías no standard Java, que permiten el acceso directo a IP [14] [15].
No sería razonable tratar de migrar la implementación presentada a espacio kernel, ya que la estructura de la misma no podría adaptarse fácilmente (además del cambio de Java a C) a operar en el entorno del sistema operativo. Sin embargo, el desarrollo y prueba rápida y sin riesgo para el sistema que posibilita esta implementación en espacio usuario, fundamentalmente de los aspectos mencionados (API y métodos de control de congestión) permite que sólo haya que modificar el kernel cuando la funcionalidad a agregar está suficientemente probada.

8. Referencias

[1] E. Kohler, M. Handley, S. Floyd, “Datagram Congestion Control Protocol (DCCP)”, RFC 4340, March 2006.
[2] http://www.ietf.org
[3] J. Postel, “Transmission Control Protocol, DARPA Internet Program Protocol Specification”, RFC 783, 1981
[4] J. Postel, “User Datagram Protocol”, RFC 768, 1980
[5] http://linux-net.osdl.org/index.php/DCCP
[6] http://www.jp.nishida.org/dccp
[7]Alkis, Evlogimenos, Khian Hao Lim, "On the Implementation of Datagram Congestion Control Protocol" 2002, www.cs.ucsd.edu/~tsohn/projects/dccp/index.html
[8] Damon Lanphear, “Datagram Congestion Control Protocol (DCCP) User Guide”, draft-ietf-dccp-user-guide-00, October 2002.
[9] T. Phelan, “Datagram Congestion Control Protocol (DCCP) User Guide”, draft-ietf-dccp-user-guide-02, July 2004
[10] Floyd, S. and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control", RFC 4341, March 2006.
[11]. M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, “TCP Selective Acknowledgment Options”, RFC 2018, October 1996
[12] Floyd, S., Kohler, E., and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)", RFC 4342, March 2006.
[13] C. Perkins, “RTP and the Datagram Congestion Control Protocol (DCCP)”, draft-ietf-dccp-rtp-07.txt, June 2007.
[14] libpcap Official Site, http://www.tcpdump.org/
[15]Jpcap Official Site, http://sourceforge.net/projects/jpcap/
[16] http://www.charles-guillemot.info/dccp.html#annexe

Improving TCP's Resistance to Blind Attacks through Ephemeral Port Randomization

Fernando Gont
Facultad Regional Haedo, Universidad Tecnológica Nacional
Haedo, Provincia de Buenos Aires, Argentina
fgont@frh.utn.edu.ar

Abstract

Recently, awareness has been raised about a number of "blind" attacks that can be performed against the Transmission Control Protocol (TCP) and similar protocols. The consequences of these attacks range from throughput-reduction to broken connections or data corruption. These attacks rely on the attacker's ability to guess or know the four-tuple (Source Address, Destination Address, Source port, Destination Port) that identifies the transport protocol instance to be attacked. While there have been a number of proposals to mitigate these Vulnerabilities, the most obvious mitigation -- TCP port randomization -- has been the one least engineered. In this paper we analyze a number of approaches for the random selection of client port numbers, such that the possibility of an attacker guessing the exact value is reduced. We discuss the potential interoperability problems that may arise from some port randomization algorithms that have been implemented in a number of popular operating systems, and propose a novel port randomization algorithm that provides the obfuscation while avoiding the interoperability problems that may be caused by other approaches. While port randomization is not a replacement for cryptographic methods, the described port number randomization algorithms provide improved security/obfuscation with very little effort and without any key management overhead.

Keywords: Transport protocols, port randomization, obfuscation, blind attacks

1 INTRODUCTION

Recently, awareness has been raised about a number of "blind" attacks that can be performed against the Transmission Control Protocol (TCP) [1] and similar protocols. The consequences of these attacks range from throughput-reduction to broken connections or data corruption [15] [12] [10]. All these attacks rely on the attacker's ability to guess or know the four-tuple (Source Address, Source port, Destination Address, Destination Port) that identifies the transport protocol instance to be attacked.

Generally, the four-tuple required to perform these attacks is not known. However, as discussed in [10] and [12], there are a number of scenarios (notably that of TCP connections established between two BGP routers [13]), in which an attacker may be able to know or guess the four-tuple that identifies a TCP connection. In such a case, if we assume the attacker knows the two systems involved in the TCP connection to be attacked, both the client-side and the server-side IP addresses could be known or be within a reasonable number of possibilities. Furthermore, as most Internet services use the so-called “well-known” ports, only the client port number might need to be guessed. Unfortunately, most systems choose the port numbers they use for outgoing connections (the so-called “ephemeral ports”) from a subset of the whole port number space, and implement ephemeral port selection algorithms that make it trivial for an attacker to guess the port numbers used by clients for outgoing connections.

In this paper we describe a method for random selection of ephemeral ports, thereby reducing the possibility of an off-path attacker guessing the exact value. This is not a replacement for cryptographic methods such as IPsec [7] or the TCP MD5 signature option [6]. However, the proposed algorithm provides improved obfuscation with very little effort and without any key management overhead.

The mechanism described is a local modification that may be incrementally deployed, and does not violate the specifications of any of the transport protocols that may benefit from it [1] [2] [8] [9].

Since the mechanism is an obfuscation technique, focus has been on a reasonable compromise between level of obfuscation and ease of implementation. Thus the algorithm must be computationally efficient, and not require substantial data structures.

2 EPHEMERAL PORTS

2.1 Traditional Ephemeral Port Number Range

The Internet Assigned Numbers Authority (IANA) assigns the unique parameters and values used in protocols developed by the Internet Engineering Task Force (IETF), including well-known ports [11]. The Internet Assigned Number Authority (IANA) has traditionally reserved the following use of the 16-bit port range of TCP and UDP:

- The Well Known Ports, 0 through 1023
- The Registered Ports, 1024 through 49151
- The Dynamic and/or Private Ports, 49152 through 65535

The range for assigned ports managed by the IANA is 0-1023, with the remainder being registered by IANA but not assigned. The ephemeral port range has traditionally consisted of the 49152-65535 range.

2.2 Traditional Ephemeral Port Selection Algorithm

As each communication instance is identified by its four-tuple {local IP address, local port, remote IP address, remote port}, selection ephemeral port numbers must result in a unique four-tuple.

TCP implementations have traditionally implemented a very simple ephemeral port selection algorithm, which simply selects ephemeral ports incrementally. Figure 1 shows the traditional ephemeral port selection algorithm in pseudocode. We will refer to this as 'Algorithm 1'.

```
next_ephemeral = 1024;  /* init., could be random */
count = max_ephemeral - min_ephemeral + 1;

do {
     port = next_ephemeral;

     if (four-tuple is unique)
         return next_ephemeral;

     if (next_ephemeral == max_ephemeral) {
         next_ephemeral = min_ephemeral;
     } else {
         next_ephemeral_port++;
     }

     count--;

} while (count>0);
```

Figure 1: Traditional ephemeral port selection algorithm

A global variable “next_ephemeral” stores the port number that should be selected the next time the port selection function is called. Table 1 shows how the algorithm could possibly select port numbers when a host establishes a number of consecutive connections to both the same and different remote sockets. From the table we note that port numbers are selected incrementally, regardless of the remote TCP socket.

Nr.	IP:port	min_ephemeral	max_ephemeral	next_ephemeral	port
#1	128.0.0.1:80	1024	65535	1024	1024
#2	128.0.0.1:80	1024	65535	1025	1025
#3	170.210.0.1:80	1024	65535	1026	1026
#4	170.210.0.1:80	1024	65535	1027	1027
#5	128.0.0.1:80	1024	65535	1028	1028

Table 1: Sample scenario for the traditional ephemeral port selection algorithm

The algorithm is simple and efficient, and is used in most TCP implementations. However, it has two weaknesses. Firstly, given that it selects port numbers incrementally and independently of the remote TCP socket, the algorithm may quickly cycle through all the port numbers in the ephemeral

port range. This may potentially lead to a port number "collision", that is, the algorithm may select a port number that results in a connection-id that is still in use in the network. Secondly, and most important for the purpose of this article, it reveals information about which port numbers will be selected for future outgoing TCP connections. If an attacker gets to know the port number selected for a recent outgoing TCP connection, he can easily guess the port numbers that will be selected for future outgoing connections.

2.3 Port Number Collisions

While it is possible for the ephemeral port selection algorithm to verify that the selected port number results in connection-id that is not currently in use at that system, there resulting connection-id may still be in use at a remote system. For example, consider a scenario in which a client establishes a TCP connection with a remote web server, and the web server performs the active close on the connection. While the state information for this connection will disappear at the client side (that is, the connection will be moved to the fictional CLOSED state), the connection-id will remain in the TIME-WAIT state at the web server for 2*MSL (Maximum Segment Lifetime). If the same client tried to create a new incarnation of the previous connection (that is, a connection with the same connection-id as the one in the TIME_WAIT state at the server), a port number "collision" would occur. The effect of these port number collisions range from connection-establishment failures to TIME-WAIT state assassination (with the potential of data corruption). In scenarios in which a specific client establishes TCP connections with a specific service at a server, these problems become evident. Therefore, an ephemeral port selection algorithm should ideally lead to a low port reuse frequency, to reduce the chances of port number collisions.

3 PORT RANDOMIZATION

As discussed in Section 1, a simple mitigation approach for all those vulnerabilities that require the attacker to guess or know the four-tuple that identifies the target connection is to obfuscate that four-tuple through a careful selection of the client port number.

There are a number of characteristics that an ideal port obfuscation algorithm should have. Firstly, it should minimize the predictability of the selected port numbers. Ideally, client port numbers should be selected randomly, and thus it would be impossible for an attacker to make an educated guess about the client port number in use by the target TCP connection.

Secondly, it should minimize the port re-use frequency, to avoid interoperability problems. A high port reuse frequency might lead to port number collisions, in which a port number is reused leading to a connection-id that is still in use in the network. These port number collisions lead to interoperability problems (the connection request will fail) which are clearly undesirable.

Finally, the port selection algorithm should avoid selecting port numbers that are needed by popular applications (such as port 80, port 6667, etc.). If a client binds a port number, and that port number is later needed by some application (while the port number is still in use), the application will fail.

3.1 Ephemeral Port Number Range

As mentioned in Section 2.1, the ephemeral port range has traditionally consisted of the 49152-65535 range. However, it should also include the range 1024-49151 range.

Since this range includes user-specific server ports, this may not always be possible, though. A possible workaround for this potential problem would be to maintain in memory an arrays of bits, in which each bit corresponds to each of the ports in the range 1024-65535. A bit set to 0 would indicate that the corresponding port is available for allocation, while a bit set to one would indicate that the port is reserved and cannot be allocated. Thus, before allocating a port, the ephemeral port selection function would check this array of bits, avoiding the allocation of ports that may be needed for specific applications.

Transport protocols should use the largest possible port range, since this improves the obfuscation provided by randomizing the ephemeral ports.

3.2 Ephemeral Port Randomization Algorithms

3.2.1 A Simple Port Randomization Algorithm

In order to address the security issues discussed in Section 2.2, a number of systems have implemented simple port number randomization algorithm, shown in Figure 2. We will refer to this algorithm as 'Algorithm 2'.

```
next_ephemeral = min_ephemeral + random()
                     % (max_ephemeral - min_ephemeral + 1)
count = max_ephemeral - min_ephemeral + 1;

do {
    if(four-tuple is unique)
            return next_ephemeral;

    if (next_ephemeral == max_ephemeral) {
        next_ephemeral = min_ephemeral;
    } else {
        next_ephemeral_port++;
    }

    count--;
} while (count > 0);

return ERROR;
```

Figure 2: Simple port randomization algorithm

This algorithm randomly selects a port number from the range {min_ephemeral, max_ephemeral} and, if the selected port number is in use, tries the next available port number (in the specified port range).

This algorithm is excellent from the point of view of obfuscation, as it selects the client port numbers randomly, making it hard for an attacker to make an educated guess about the client port number in use for the target TCP connection. However, it has a number of weaknesses.

Since this algorithm performs a completely random port selection (i.e., without taking into account the port numbers previously chosen), it has the potential of reusing port numbers too quickly. Consequently multiple ports may have to be tried and verified against all existing connections before a port can be chosen. Although carefully chosen random sources and optimized four-tuple lookup mechanisms (e.g., optimized through hashing), will mitigate the cost of this verification, some systems may still not want to incur this unknown search time.

Additionally, potentially high port reuse frequency might lead to port number collisions at the server side, which would lead to the interoperability problems discussed in Section 3.2 of this paper. Systems that may be specially susceptible to this kind of repeated four-tuple collisions are those that create many connections from a single local IP address to a single service (i.e. both IP addresses and server port are fixed). Gateways such as proxy servers are an example of such a system.

3.2.2 A novel port obfuscation algorithm

Figure 3 shows the pseudocode for a novel port obfuscation algorithm [14], modeled after the ISN (Initial Sequence Number) selection algorithm described in RFC 1948 (“Defending Against Sequence Numbers Attacks”). The algorithm aims to achieve the obfuscation quality of the simple port randomization algorithm described in the previous section, while keeping the port reuse frequency properties of the traditional TCP port selection algorithm.

```
next_ephemeral = 1024;  /*init., could be random */

offset = F(local_IP, remote_IP, remote_port, secret_key);

do {
    port = min_ephemeral + (next_ephemeral + offset)
               % (max_ephemeral - min_ephemeral + 1);
    next_ephemeral++;

    if(four-tuple is unique)
        return port;

    count--;

} while(count > 0);

return ERROR;
```

Figure 3: A Novel Port Obfuscation Algorithm

The strategy to achieve both goals is to separate the port number space for each remote TCP socket, producing a monotonically-increasing port number sequence (with a random initial port number) for each of them. That is, two consecutive connection requests sent to different TCP sockets would use unrelated client port numbers, while two consecutive connection requests to the same TCP endpoint would use incremental port numbers.

Ephemeral port numbers are selected as the sum of the result of a function F() and the variable “next_ephemeral”. F() is a hash function fed with the server TCP socket {server IP address, server TCP port} and a secret key specified by the system administrator or randomly chosen at system startup. This function F() provides a random “offset” that will be different for each remote TCP

socket. On the other hand, “next_ephemeral” is incremented each time the algorithm selects an ephemeral port, thus ensuring that two consecutive outgoing connections will use different ephemeral port numbers.

Thus, F() provides for the obfuscation in the port number selection, while “next_ephemeral” provides for a monotonically-increasing sequence. Provided F() is a cryptographically-secure hash function, and that the attacker does not know the secret key used as input to F(), it will be very difficult for an off-path attacker to guess the ephemeral port number selected for the connection.

Ideally, the algorithm would use one “next_ephemeral” variable for each of the possible results of F(). However, as this would require a probably unacceptable amount of memory, the algorithm uses a single global “next_ephemeral” variable. The downside of this engineering decision is that the selection of a port number in any port number sequence will cause all the other port number sequences to “skip” a port number they could have potentially used, unnecessarily.

An analysis of a sample scenario can help to understand how this algorithm works. For example, let’s suppose that some host tries to establish TCP connections with a number of remote TCP sockets. Table 2 illustrates, for a number of consecutive connection requests, some possible values for each of the variables used in this novel port obfuscation selection algorithm. Additionally, the table shows the result of the port selection function.

Nr.	IP:port	offset	min_ephemeral	max_ephemeral	next_ephemeral	port
#1	128.0.0.1:80	1000	1024	65535	1024	3048
#2	128.0.0.1:80	1000	1024	65535	1025	3049
#3	170.210.0.1:80	4500	1024	65535	1026	6550
#4	170.210.0.1:80	4500	1024	65535	1027	6551
#5	128.0.0.1:80	1000	1024	65535	1028	3052

Table 2: Sample scenario for the novel port obfuscation algorithm

The first two entries of the table illustrate the contents of each of the variables when two ephemeral ports are selected to establish two consecutive connections to the same remote socket {128.0.0.1, 80}. We can see that the two ephemeral ports that get selected belong to the same port number “sequence”, as the result of the hash function F() is the same in these two cases.

The second and third entries of the table illustrate the contents of each of the variables when the algorithm later selects two ephemeral ports to establish two consecutive connections to the remote socket {172.0.0.1, 110}. We can see that the result of F() is the same for these two cases, and thus the two ephemeral ports that get selected belong to the same “sequence”. However, this sequence is different from that of the first two port numbers selected before, as the value of F() is different from the one obtained for those two ports numbers (#1 and #2) selected earlier.

Finally, when the algorithm later selects another ephemeral port to connect to the same socket as in #1 and #2, we note that the selected port number somehow belongs to the same sequence as the first two port numbers selected (#1 and #2), but that two ports of that sequence (3050 and 3051) have been skipped. This is the consequence of having a single global next_ephemeral variable that gets incremented whenever a port number is selected. When next_ephemeral is incremented as a result of the port selections #3 and #4, this causes two ports (3050 and 3051) in all the other the port number sequences (including that of #1 and #2) to be “skipped”, unnecessarily.

As in the case of the traditional TCP port selection algorithm, having a single global counter for the port numbers that have so far been selected may result in a port reuse frequency higher than needed.

The obvious mitigation for this effect would be to have a different “next_ephemeral” variable for each possible result of F(). Thus, assuming no hash collisions, the selection of a port number would increment only the corresponding next_ephemeral variable, without causing port numbers in other sequences to be skipped. However, this would likely require an unacceptable amount of memory.

A middle-ground between a single global next_ephemeral variable and a large number of next_ephemeral variables (one for each possible result of F()) would be to have a small number of next_ephemeral variables, such that each possible value of F() is matched to one of these variables. For example, we could define an array of 256 variables, each of them representing a different next_ephemeral variable. The index into this array could be the result of a hash function G() computed with the remote TCP socket {remote IP address, remote TCP port} and a secret key, or even a value derived from the result of F() (for example, the result of performing an eXclusive-OR among each of the bytes composing the result of F(). Figure 3 shows the pseudocode for the improved algorithm.

```
/* Initialization code */
for(i = 0; i < TABLE_LENGTH; i++)
    table[i] = random % 65536;


/* Ephemeral port selection */
offset = F(local_IP, remote_IP, remote_port, secret_key);
index = G(offset);
count = max_ephemeral - min_ephemeral + 1;

do {
    port = min_ephemeral + (offset + table[index])
               % (max_ephemeral - min_ephemeral + 1);

    table[index]++;
    count--;

      if(four-tuple is unique)
    return port;

} while (count > 0);

return ERROR;
```

Figure 4: An improvement to the novel port obfuscation algorithm

In order to illustrate how this slight modification can improve the port reuse properties of the novel port obfuscation algorithm discussed earlier, we can analyze the scenario of the previous section, this time from the perspective of the improved algorithm. For the purposes of illustrating how the improved algorithm works, we will refer to the array of next_ephemeral variables as “table”, and for simplicity-sake we will assume that all the entries of the array have been initialized to 1024. Also, we will use the variable “index” to store the value used as the index into the array “table”.

Table 3 illustrates a possible result for the same sequence of events as those in table 2, along with the values of each of the involved variables.

Nr.	IP:port	offset	min_ephemeral	max_ephemeral	index	table[index]	port
#1	128.0.0.1:80	1000	1024	65535	10	1024	3048
#2	128.0.0.1:80	1000	1024	65535	10	1025	3049
#3	170.210.0.1:80	4500	1024	65535	15	1024	6548
#4	170.210.0.1:80	4500	1024	65535	15	1025	6549
#5	128.0.0.1:80	1000	1024	65535	10	1026	3050

Table 3: Sample scenario for the improved novel port obfuscation algorithm

From the table we can see that the destinations “128.0.0.1:80” and “170.210.0.1:80” result in different values for “index” and, as a result, our slight modification successfully avoids the increments in one of the port number sequences to affect the other sequences, thus minimizing the port reuse frequency.

3.2.3 Secret Key

Every complex manipulation (like MD5) is no more secure than the input values, and in the case of ephemeral ports, the secret key. If an attacker is aware of which cryptographic hash function is being used by the victim (which we should expect), and the attacker can obtain enough material (e.g. ephemeral ports chosen by the victim), the attacker might simply search the entire secret key space to find matches.

To protect against this, the secret key should be of a reasonable length. Key-lengths of 32-bits should be adequate, since a 32-bit secret would result in approximately 65k possible secrets if the attacker is able to obtain a single ephemeral port (assuming a good hash function). If the attacker is able to obtain more ephemeral ports, key-lengths of 64-bits or more should be used.

Another possible mechanism for protecting the secret key is to change it after some time. If the host platform is capable of producing reasonable good random data, the secret key can be changed. Changing the secret will cause abrupt shifts in the chosen ephemeral ports, and consequently collisions may occur. Thus the change in secret key should be done with consideration and could be performed whenever one of the following events occur:

- Some predefined/random time has expired
- The secret has been used N times (i.e. we consider it insecure).
- There are few active connections (i.e., possibility of collision is low).
- There is little traffic (the performance overhead of collisions is tolerated).
- There is enough random data available to change the secret key (pseudo-random changes should not be done).

4 CONCLUSIONS

Ephemeral port randomization can provide efficient mitigation for blind attacks against transport protocols, without any key management overhead. Implementation and deployment experience have shown that trivial port randomization approaches can lead to interoperability problems. In this paper we have examined a novel algorithm to achieve port number obfuscation, while avoiding the potential interoperability problems introduced by other alternative approaches.

4 ACKNOWLEDGEMENTS

The author would like to thank Guillermo Gont and Juan Fraschini for reviewing a draft version of this paper. The author would also like to thank FreeBSD's Mike Silbersack for a very fruitful discussion about ephemeral port selection techniques

5 REFERENCES

[1] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[2] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[3] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[4] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[5] Bellovin, S., "Defending Against Sequence Number Attacks", RFC 1948, May 1996.

[6] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[7] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[8] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[9] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[10] Watson, P., "Slipping in the Window: TCP Reset attacks", December 2003.

[11] “IANA Port Numbers”, <http://www.iana.org/assignments/port-numbers>.

[12] Touch, J., "Defending TCP Against Spoofing Attacks", draft-ietf-tcpm-tcp-antispoof-05 (work in progress), October 2006.

[13] Rekhter, Y., Li, T., Hares, S., “A Border Gateway Protocol 4 (BGP-4)”, RFC 4271, January 2006.

[14] Larsen, M., Gont, F., "Port Randomization", draft-larsen-tsvwg-port-randomization-01.txt (work in progress), Feb. 2007.

[15] Gont, F., "ICMP attacks against TCP", draft-ietf-tcpm-icmp-attacks-01 (work in progress), October 2006.

Integración de herramientas par la Simulación y modelado de redes WAN en el desarrollo de aplicaciones distribuidas

Orlando Micolini
Laboratorio de Arquitectura de Computadoras, FCEFyN
Universidad Nacional de Córdoba
Córdoba, Argentina
lac@efn.uncor.edu

Adriana Damiani
Laboratorio de Arquitectura de Computadoras, FCEFyN
Universidad Nacional de Córdoba
Córdoba, Argentina
adamiani@efn.uncor.edu

Abstract

Actually there is a great development interest and use of distributed system. In those systems the communication network has the most important role in its operation. The complexity of setting up these systems (effort, time, and money) is strongly influenced by different parameters and network configurations. This takes us to propose an instrument for the net simulation of a distributed system.
In this paper we present a multi-platform tool, implemented with a parallel architecture, for the evaluation and testing of the parameters in the net for tuning system performance.
Our purpose constitutes an integration and extension of existing software for the net simulation. This tool allows the messages capture between subsystems, the injection of these messages into the simulator and its recovery to send them to the target subsystem.

Keywords: Testing Distributed Systems, Simulation, Communications and Networks, Parallel Software.

Resumen

Actualmente hay gran interés en desarrollo y uso de sistemas distribuidos. En estos, la red de comunicaciones cumple un rol fundamental para su funcionamiento. La complejidad de la puesta a punto (esfuerzo, tiempo y dinero) de dichos sistema esta fuertemente influenciada por los distintos parámetros y configuraciones de la red. Esto nos lleva a proponer un instrumento para la simulación de la red de un sistema distribuido.
En este trabajo presentamos una herramienta multi-plataforma, implementada con una arquitectura paralela, para la evaluación de la influencia de los parámetros de la red en las prestaciones del sistema. Nuestra propuesta constituye una integración y extensión de software existentes para la simulación de redes. Esta herramienta permite la captura de mensajes entre susbsistemas, la inyección de estos mensajes al simulador y su recuperación para enviarlos al subsistema destino.

Palabras Claves: Prueba de sistemas distribuidos, Simulación, Comunicaciones y Redes, Software Paralelo.

1 INTRODUCCIÓN

El presente trabajo aborda los requerimientos y la implementación de una herramienta para el desarrollo, prueba y validación de aplicaciones en sistemas distribuidos. El comportamiento de sistemas distribuidos es más complejo que los sistemas que se ejecutan en un único nodo, puesto que una simple operación puede involucrar a cientos de nodos y el envío y recepción de miles de mensajes, con el agravante de que los componentes se encuentran distribuidos geográficamente.
Por ejemplo, si queremos determinar el comportamiento de una implementación frente a la perdida de datos y o variaciones de los retardos, se hace difícil realizar y observar el experimento debido a la distribución y a la variabilidad de comportamiento de la red WAN.
Para esto hemos implementado una herramienta, que permita realizar el experimento en una LAN y arroje los resultados del sistema en evaluación para su análisis, donde los tiempos de ejecución son similares a los del sistema real.

1.1 Objetivos

Principalmente se pretende:

- Modelar una red de comunicaciones de datos para aplicaciones distribuidas, flexible en su topología y que permitan recolectar el contenido de los paquetes para su posterior análisis
- El Simulador de red permite que los programas a probar corran sin modificaciones en la implementación ni el ambiente de trabajo. El sistema real puede acoplarse al simulador sin modificaciones.
    - Inyección de Fallos como perdida de información y variación en los retardos, para evaluar las soluciones de sistemas distribuidos

Como objetivos secundarios: el simulador esta paralelizado, permitiendo su implementación en sistemas distribuidos, tanto débilmente acoplados como fuertemente acoplados.

2 EL CONTEXTO

Existen muchas herramientas complementarias para la detección de errores e inyección de errores en sistemas distribuidos.
Algunas de las herramientas mas usadas son: gdb y gprof para bugs de bajo nivel y un nodo, caja negra [1] para sistemas los que no se tiene el código fuente, impresiones y asserts para la detección de errores por análisis de log, model checking [2], herramientas basadas en agentes [3] para el manejo y o recolección de path event e información del sistema.
Ninguna de estas herramientas permite la realizar de la evaluación del sistema distribuido mientras varían los parámetros de la red y sin ningún cambio de configuración o código.

3 REQUERIMIENTOS

Para lograr una arquitectura adecuada, es necesario explicitar los requerimientos. Dichos requerimientos están implícitos en los objetivos [16].
Para la simulación de la red, se deben cumplir con las siguientes condiciones:

1. Implementación de la red por segmentos independientes con el fin de distribuirla en distintos procesadores
2. Mecanismo de comunicación estándar que permita el acoplamiento, tanto a otros segmentos de red, como a los sistemas que se quiere evaluar
3. Debe ser multi-plataforma, puesto que esta herramienta puede requerir múltiples nodos con diferentes SO
4. La herramienta tiene que ejecutarse en la maquina local, adonde se prueba la aplicación sin modificar
5. La herramienta tiene que poder acoplarse a las aplicaciones distribuidas, capturando los mensajes. Esto se realizara por TCP/IP

4 ARQUITECTURA

En la figura 1, los subsistemas 1 y 2 conforman el sistema distribuido a evaluar. Estos se comunican a través de una red, la que puede ser LAN o WAN.
La herramienta propuesta es un sistema que realiza:

- La intercepción de los paquetes que el **Subsistema1** envía al **Subsistema2**
- Inyecta los paquetes en una red simulada
- Recupera los paquetes y los inyecta en el **Subsistema2 (**Figura 1)

De esta forma no se requieren cambios en los subsistemas como recopilación de código y/o cambios de configuración.

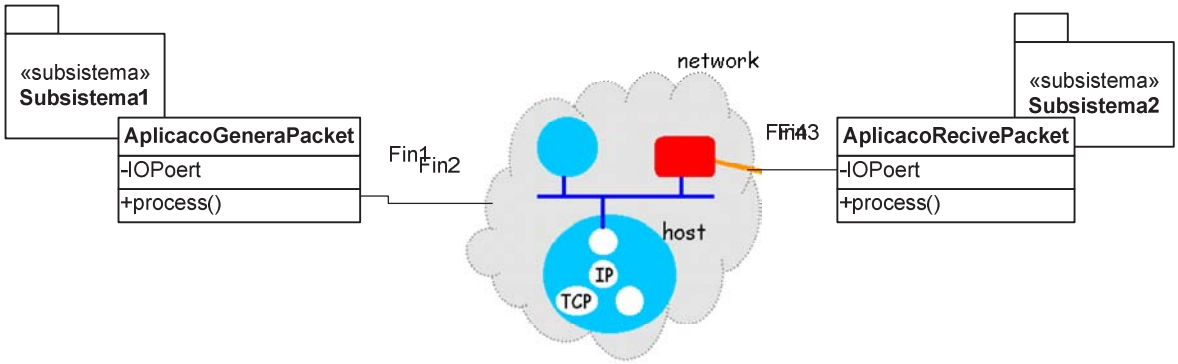


Figura 1

Con respecto a la red simulada, se tiene control de sus paramentos y topología, por lo que es posible extenderla y/o paralelizarla. Esto constituye el aporte fundamental de la herramienta desarrollada.
De la Figura 1 y de los requerimientos se desprende que las partes principales del sistema son:

1. Modulo de captura: captura los mensajes de la aplicación fuente (paquetes) y los envía al simulador
2. Modulo de simulación: es programable según la topología y los parámetros de la red. Es divisible en partes con el fin poder ejecutarlos en distintos procesadores
3. Modulo de transmisión: extrae los mensajes (paquete) del simulador, reconstituye el mensaje y lo envía al subsistema destino
4. Modulo de almacenamiento: almacena la información contenida en los mensajes (paquetes de comunicación) con el fin de recuperarlos para su análisis posterior, y recuperación de los mensajes
5. Modulo de configuración y gestión: permite la configuración del sistema y de la topología de la red, las direcciones de los host y la distribución de los procesos de simulación y captura

Todos estos componentes, mostrados en la figura 2, se tienen que poder ejecutar en múltiples plataformas sin modificación.

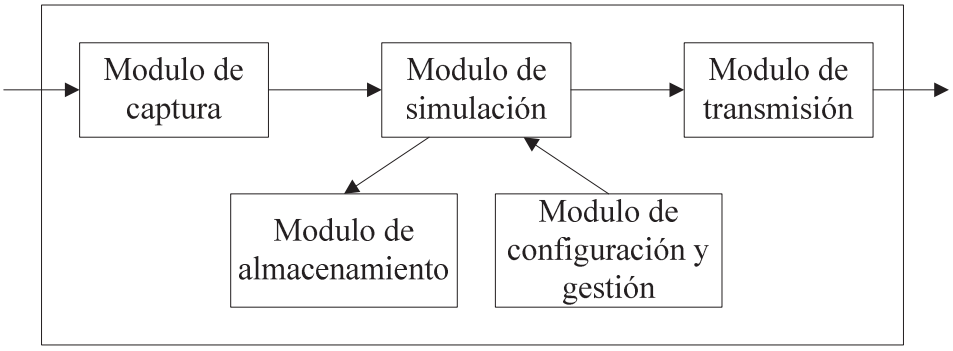


Figura 2: Arquitectura del sistema

5 ANÁLISIS DE HERRAMIENTAS

5.1 Modulo de Simulación

Se realizo una búsqueda de software de simulación orientado a redes, según los requerimientos expuestos. En primer lugar se enumeran algunas de las herramientas mas usadas y probadas, seguidamente se selecciono la que mas se adecue a nuestro proyecto.

Nombre	Network Simulator Tesbed (NEST) [4]
Desarrollado por	Computer Science de la Columbia University
Aplicación	Ambiente de simulación para sistemas de redes distribuidas y protocolos básicos
Implementado	En C y permite a los usuarios ejecutar sus propios códigos en C
Otras características	Interfaz grafica que permite controlar la simulación Basado en una arquitectura cliente/servidor, lo que permite simular complejos escenarios.

Nombre	Maryland Routing Simulator (MaRS) [6]
Desarrollado por	Computer Science de la University of Maryland
Aplicación	Estudio de algoritmos de ruteo en redes WAN
Implementado	En C en una plataforma Unix
Otras características	Es una evolución de un simulador mas antiguo llamado NetSim

Nombre	Realistic And Large Network Simulator (Real) [12]
Desarrollado por	Hani T. Jamjoom de Cornell University
Aplicación	Estudio del comportamiento dinámico de flujos y de control de

	congestión en redes de datos packet switched
Implementado	En C y también se ha desarrollado una interfaz de usuario grafica (GUI) en Java
Otras características	Estudio de sistemas o parámetros que no afecten de forma directa al flujo de conexiones TPC/IP principal. Esto limita la capacidad de modelar un sistema real

Nombre	Network Simulator 2 (ns-2) [7]
Desarrollado por	DARPA, la Universidad del Sur de California y patrocinado por Xerox
Aplicación	
Implementado	En C++, pero para realizar las simulaciones usa un lenguaje interpretado llamado Tcl
Otras características	Es una evolución de REAL network Simulator.Posee capacidades de routing y multicast en redes estructuradas y gíreles. Posee un visualizador llamado Nam, que permite ver en forma simple los resultados de la simulación

Nombre	S3 project, Scalable Simulation Framework [8]
Desarrollado por	Renesys Corporation, el Institute for Security Technology Studies at Dartmouth y DARPA
Aplicación	Permite el uso de prácticamente todos los protocolos en Internet.
Implementado	En C++ y posee dos interfaces de programación, una en Java y otra C++, así como en DML
Otras características	Es altamente escalable pero tiene escaso rendimiento en las versiones gratuitas. La interacción con la simulación solo puede hacerse a través del DML

Nombre	J-Sim Java Simulator [5]
Desarrollado por	NSF, DARPA, CISCO y las Universidades de Illinois y Ohio
Aplicación	
Implementado	En Java, basado en el modelo de programación de componentes autónomos
Otras características	Posee una interfaz de script para la integración con lenguajes de script como Perl, Tcl o Pitón Soporta la mayoría de los protocolos usados en Internet, así como multicast y QoS.

Nombre	Gíreles IP Simulator (WIPSIM) [9]
Desarrollado por	Departament of Communication Technology de la Aalborg University (Dinamarca), en curso
Aplicación	Simulación de redes wireless en IPv6. Su uso principal es en investigación y estudio de mecanismos de control de congestión, escenarios móviles, protocolos y descubrimientos de rutas.
Implementado	En C++.
Otras características	Es fácil escribir protocolos en las capas de transporte, red, link y MAC, y tiene implementaciones para UDP, DiffServ, ISMA, CSMA y Blue Tooth.

El software de simulación elegido fue J-Sim debido a que soporta tanto múltiples plataformas y permite la descomposición para distribuir las cargas de simulación en múltiples procesadores. La implementación de simulador de res se hace por componentes. Esto se logra debido a su arquitectura orientada a componentes [17].

5.2 Modulo de Captura

El objetivo es el diseño e implementación del modulo de acoplamiento de recepción y transmisión de captura de mensajes en el sistema distribuido a evaluar.
La comunicación entre procesos se basa en un Middleware, que esta conformado por diferentes capas. Dichas capas pueden implementar cualquiera de los paradigmas de programación distribuida. En su mayoría están soportados por los protocolos UDP y TCP [13].
Por ejemplo:

	Aplicaciones Servicios	
	RMI y RPC	Capa Middleware
	Protocolo Petición-Respuesta Empaquetado y representación externa de Datos	
	UDP y TCP	

RMI invocación de métodos remotos, permite que un objeto invoque métodos de otro objeto remoto (ej. CORBA y Java RMI).
RPC llamada a procedimientos remotos, permite que un cliente ejecute procedimiento de un servidor remoto.
La capa que da servicio para comunicación Petición-Respuesta entre procesos, valiéndose de TCP, UDP. Los cuales están implementados con sockets tanto en Java como en UNIX.

5.2.1 Características de la Comunicación entre Procesos

El proceso emisor envía un mensaje al proceso receptor, esto implica la comunicación de datos y también puede implicar la sincronización entre procesos.
Las operaciones básicas son *send* y *receive*. Como se puede ver en el esquema de capas, estas están soportadas por protocolos de comunicaciones, principalmente por paquetes TCP y UDP, según sean sincrónicas o asincrónicas.
Lo antedicho implica que nuestro modulo tiene que ser capaz de capturar y filtrar paquetes UDP y TCP para inyectarlos al modulo de simulación. También tiene que poder identificar sus destinos, para rutearlos dentro del simulador y al subsistema de destino. Para esto se deben capturar otros tipos de paquetes como ARP, RARP con el fin de determinar direcciones IP de los distintos [14].
De lo expuesto anteriormente se desprende que el manejo de sockets no es suficiente y se requiere una librería que permita acceder tanto a los datos de los paquetes como a la información de sus encabezados.
Uno de los modulo mas usados y probados para la captura de paquetes es PCap, pero esta desarrollado en C, por lo que debe ser recompilado para distintas plataformas. La versión Java, que nos permite ejecutarlo en múltiples plataformas, es JPcap [10], y es la que mas se adecua a nuestro proyecto, por lo que se eligió el JPcap para el desarrollo del modulo de captura.

5.3 Modulo de Configuración, Almacenamiento y Gestión

El modulo de configuración realiza la persistencia de:

a. La topología de la red, la subdivisión en segmentos y los procesos para ser ejecutados en los distintos hosts
b. Los puertos y sus direcciones para realizar la comunicación con los subsistemas en evaluación
c. Los mensajes contenidos en los paquetes de comunicación para su posterior recuperación

Estos procesos estarán distribuidos entre los hosts que conforman la herramienta y se deben comunicar y sincronizar con el fin de realizar la simulación y captura de datos.
Siendo que tanto los módulos de simulación como de captura están implementados en java, aquí también se realizo la implementación en java.

6 LA SOLUCIÓN QUE SE PROPONE

Para la comunicación y sincronización de los mensajes entre los distintos módulos de la herramienta hay dos alternativas claras que son socket y RMI [11] [15]:
Se decidió hacer la primera la primera aproximación con RMI, con el fin de mantener un diseño de objetos.
En ambas alternativas, hay que serializar [15] los objetos para almacenarlos y transmitirlos entre procesos, estos son los packetes capturados UDP, TCP, ARP y RARP. Lo importante a tener en cuenta

es que los paquetes capturados por j-pcap no son serializables, para lo que se deben discriminar y extraer la información para crear un nuevo objeto paquete serializable. Se procesa dicha información para luego reconstituir el paquete que se envía al proceso destino.

6.1 Implementación del Modulo de Recepción y Transmisión

6.1.1 Modulo de Recepción

En la figura 3 se muestra el diagrama de objetos de este modulo. El objeto CapturePacketNIC implementa PacketReceiver (del J-pcap); el JpcapCaptor captura los paquetes y los manda a este objeto. CapturePacketNIC contiene una referencia a un objeto del tipo ComNICtoRMI. Dicho objeto realiza la discriminación de paquetes y los convierte a objetos serializable para poder hacer la comunicación interproceso por RMI

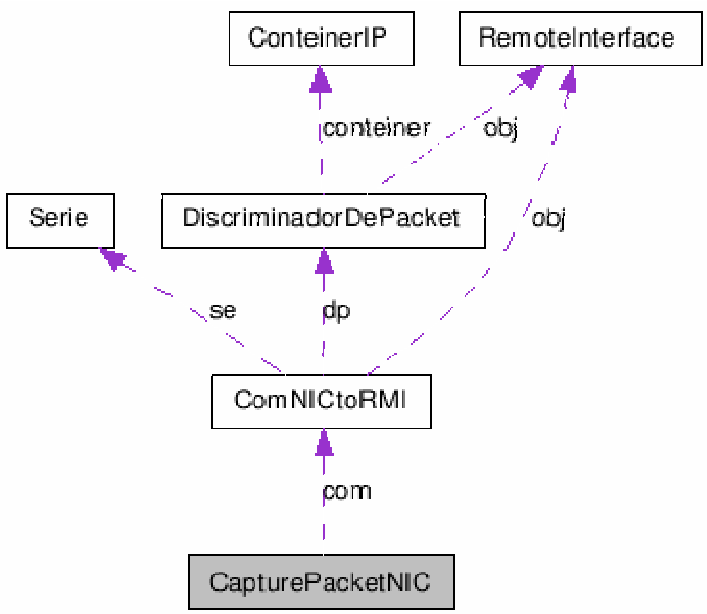


Figura 3: Diagrama de clases del modulo de recepción

El discriminador de paquetes extrae los paquetes ARP para armar las tablas de ruteo y los paquetes UDP/TCP de donde saca la dirección de destino para inyectarlos a la red simulada y al container para su almacenamiento.

La secuencia de flujo de mensajes del modulo de recepción se muestra en la figura 4.

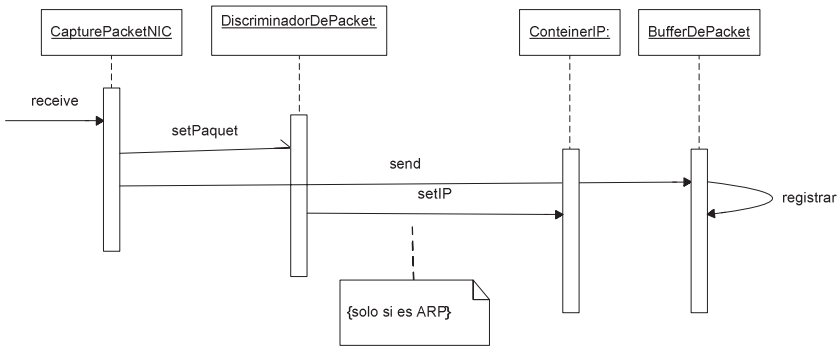


Figura 4: Diagrama de secuencia del modulo de recepción

6.1.2 Modulo de Transmisión

Este modulo es invocado por el simulador cuando un paquete sale del mismo y debe ser inyectado a la red para alcanzar el subsistema de destino.
La clase ContinerIP mantiene una tabla con las relaciones IP destino a NIC del host.
La clase Send se encarga de recuperar el paquete de BufferDePacket (contenedor de los paquetes). En la figura 5 se muestra el diagrama de clases de este modulo.

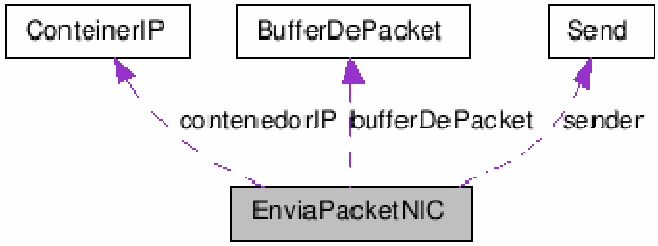


Figura 5: Diagrama de clases de Modulo de Transmisión

6.2 Implementación del Modulo de Simulación

Para el modulo de simulación se utilizo J-Sim. Con el fin de realizar la comunicación se crearon las siguientes clases: GenPacket y RecvPacket. Ambas extienden Component de JSim y poseen puertos de entrada salida para inyectar/recibir paquetes.

En la figura 6 se muestra una configuración elemental del simulador, la cual puede ser escalada en distintas topologías de red.

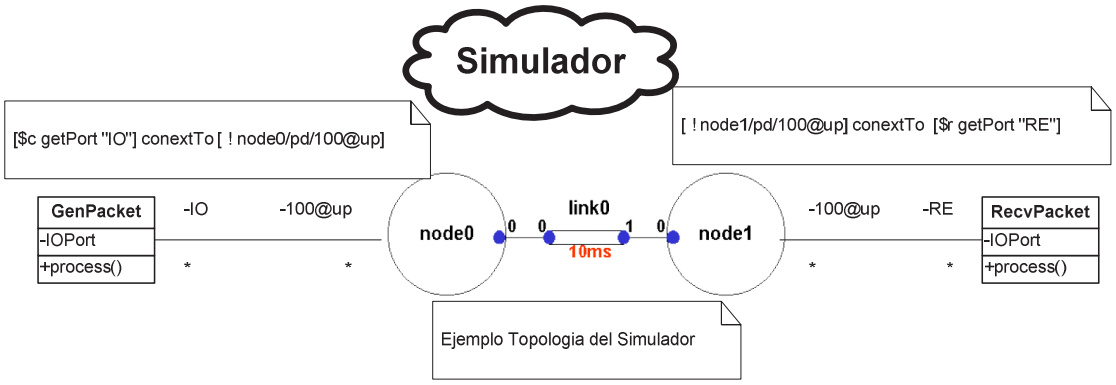


Figura 6: Diagrama de acoplamiento del Simulador

6.3 Implementación del Modulo de Almacenamiento

El objeto BufferDePacket (figura 5) realiza la persistencia de los paquetes que conforman el tráfico del simulador. Cuando el simulador determina que un paquete arribo a su destino, RecvPacket se lo comunica a EnviaPacketNIC, quien lo recupera de BufferDePacket y le coloca una marca de tempo. Para luego enviarlo a un buffer de persistencia definitivo.

7 Conclusión

La solución propuesta se adecua a los requerimientos planteados: multiplataforma, no requiere cambios del sistema a evaluar y permite distintas topologías de red.
El hecho de implementar el simulador por componentes simplifica la paralelización de este modulo.
Una de las principales fortalezas de la herramienta es su escalabilidad. Al dividir y desacoplar la implementación en captura y simulación, es posible hacer uso de un procesador para realizar esta la simulación y que varios procesadores realicen la captura. Esto es recomendable puesto que la captura se debe hacer en tiempo real.

7.1 Mejoras

Realizar un plug-in para herramientas como eclipse con el fin de integrarlo a un framework para así darle al conjunto una mayor funcionalidad.
Implementar una interfase grafica para la configuración y la creación de la topología de la red simulada. Implementar un visualizador grafico para los resultados.

Bibliografía

[1] M. K. Aguilera, J. C. Mogul, J. L. Wiener, P. Reynolds, and A. Muthitacharoen. Performance debugging for distributed systems of black boxes. In Proc. SOSP, Bolton Landing, NY, Oct. 2003.
[2] P. Godefroid. Software model checking: the VeriSoft approach. FormalMethods in System Design, 26(2):77–101, Mar. 2005.
[3] M. K. Aguilera, J. C. Mogul, J. L. Wiener, P. Reynolds, and A. Muthitacharoen. Performance debugging for distributed systems of black boxes. In Proc. SOSP, Bolton Landing, NY, Oct. 2003
[4] Sitio oficial de real. http://www.cs.cornell.edu/skeshav/real/overview.html
[5] Sitio oficial de j-sim. http://www.j-sim.org/.
[6] Sitio oficial de MaRS http://www.ccs.neu.edu/home/matta/software.html
[7] Sitio oficial de ns-2. http://www.isi.edu/nsnam/ns/.
[8] Sitio oficial de s3. http://dimacs.rutgers.edu/Projects/Simulations/darpa/.
[9] Sitio oficial de wipsim. http://sourceforge.net/projects/wipsim/.
[10] Sitio oficial de Jpcap. http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html
[11]Sitio oficial de Sun, RMI http://java.sun.com/docs/books/tutorial/rmi/.
[12] Sitio oficial de nctuns. http://nsl10.csie.nctu.edu.tw/.
[13] George Coulouris, Jean Dollimore Y Tim Kindberg, ``Sistemas Distribuidos. Coceptos Y Diseño". 3ª Edición. Pearson Educación, 2001.
[14] W. Richard Stevens, The Protocols TCP/IP Illustrated, Volume 1, Addison-Wesley, 1994
[15] William Grosso, Java RMI, Publisher: O'Reilly, First Edition October 2001
[16] SOMMERVILLE IAN, INGENIERIA DEL SOFTWARE ,PEARSON EDUCACION, Edición 2005
[17] Clemens Szyperski , Component Software, Addison-Wesley Professional, 1997

INTERFACE PCI PARA PC COMPATIBLE

Ing. Luis O. Ventre - Ing. Leonardo Gamarra
Laboratory of Computer Architecture - Universidad Nacional De Córdoba
Córdoba, C.P. 5000 – Argentina
lventre@gmail.com - leogamarra@gmail.com

Supported By:
Ing. Orlando Micolini – Ing. Adriana Damiani
Laboratory of Computer Architecture, Universidad Nacional de Córdoba
Córdoba- CP 5000 –Argentina
omicolini@compuar.com - adriana.damiani@gmail.com

ABSTRACT:

We call BUS to any group of lines whose main purpose its interconnect different devices. One of the most critical problem in digital aplicattions its to establish a fast access communication between peripheral devices. With the intention of diminishing the limitations in the data transfer rate, in year 1992, Intel develops the Standard bus of interconnection of peripheral components (PCI).

The PCI Bus, is the most used standard for the development of aplication where work togheter external components to the CPU; this paper discuss the development of my end career project, which raises the implementation of a CORE, intrument that allows to fulfill the highest requirements of the communication protocol PCI, this instrument is implemented on a reprogrammable logic FPGA Board.

Also, explain the implement of drivers for testing the development under an operating system, and several visual aplications were created to validate the communication and bidirectional data transfer.

At the end of this paper, we measured and studied the results to specify the reasons that allows us to affirm that the development is successful.

Keywords: Computer Architecture, PCI Bus, Core, FPGA.

1-INTRODUCCIÓN:

Los buses son los encargados de realizar el intercambio de información entre dispositivos periféricos y la CPU en la arquitectura de computadoras, su rol es fundamental en todo tipo de aplicaciones. Hasta hace un tiempo los principales buses como ISA, EISA o Micro Channel presentaban, ante aplicaciones gráficas, o de requerimientos elevados, problemas bien determinados como:

- Baja frecuencia de operación.
- Longitud de palabra limitada.
- Importante latencia de acceso a los dispositivos.
- Poca escalabilidad.

Con el avance de la tecnología de los periféricos multimedia y de almacenamiento masivo de información, creció significativamente la necesidad de intercambiar grandes cantidades de datos, con la consiguiente necesidad de aumentar la velocidad de transferencia de los mismos. Estos problemas provocaron una inevitable evolución en los buses con ideales óptimos de:

- Flexibilidad – Amplio espectro de conexión a distintos dispositivos.
- Alcanzar un nivel de estándar a nivel de mercado mundial.
- Expansibilidad – para la admisión de numerosos dispositivos.
- Performance - alta velocidad de interconexión .

Con esta perspectiva surge en el año 1992, desarrollado por Intel la norma que define al Bus PCI (Peripheral Component Interconnect), PCI (PCI-SIG) [1] define, "*el objetivo es desarrollar un estándar industrial de la arquitectura de bus local de altas prestaciones que facilite el desarrollo de nuevos periféricos"*

Bajo esta arquitectura de Bus es factible realizar incontadas aplicaciones con diversas funcionalidades, pero todas y cada una de ellas requiere el desarrollo de un CORE PCI, el cual es el núcleo que implementa el protocolo de intercomunicación tal cual define la norma. Debido a los requisitos de la misma, la implementación de dicho CORE necesita de manera indispensable de lógica avanzada, como por Ej.: FPGA o chips PLX[2].
En el presente trabajo se desarrolla un CORE PCI, en una FPGA y se implementan aplicaciones de transferencia de datos.

2-REQUERIMIENTOS

- Desarrollar el protocolo de comunicación para lograr la interconexión bidireccional entre un dispositivo esclavo y un maestro a través del bus PCI.
- Implementarlo en un dispositivo esclavo.
- Desarrollar un driver para poder tener acceso al dispositivo y a los registros de configuración del CORE PCI, con el fin de realizar el testing.
- Realizar una o mas aplicaciones para demostrar la bidireccionalidad de la comunicación entre los dispositivos.

[1] PCI (PCI-Special Interest Group) www.pcisig.com - [PciSpec, 98]
[2] PLX Technology www.plxtech.com

3-OBJETIVOS

Realizar el desarrollo de un CORE que implemente la interfaz PCI en una FPGA permitiendo minimizar los problemas descriptos en la introducción. Para cumplir este objetivo es necesario:

- Seleccionar el Kit Fpga para la implementación.
- Elección del lenguaje de programación para el desarrollo.
- Desarrollar el CORE con las herramientas selectas.
- Realizar uno o mas Drivers para el testing bajo sistemas operativos.
- Realizar aplicaciones y hard necesarios para validar el funcionamiento del desarrollo.

4-DESARROLLO

4.1-Nuestro proyecto

El lenguaje selecto para el desarrollo fue VHDL (Very High Speed Integrated Circuit Hardware Description Language) por sus siguientes ventajas:

- Standard principal de mercado.
- Primordial lenguaje de desarrollo de aplicaciones PCI.
- Alta disponibilidad de recursos bibliográficos.

El desarrollo implementado puede observarse en la figura 1:

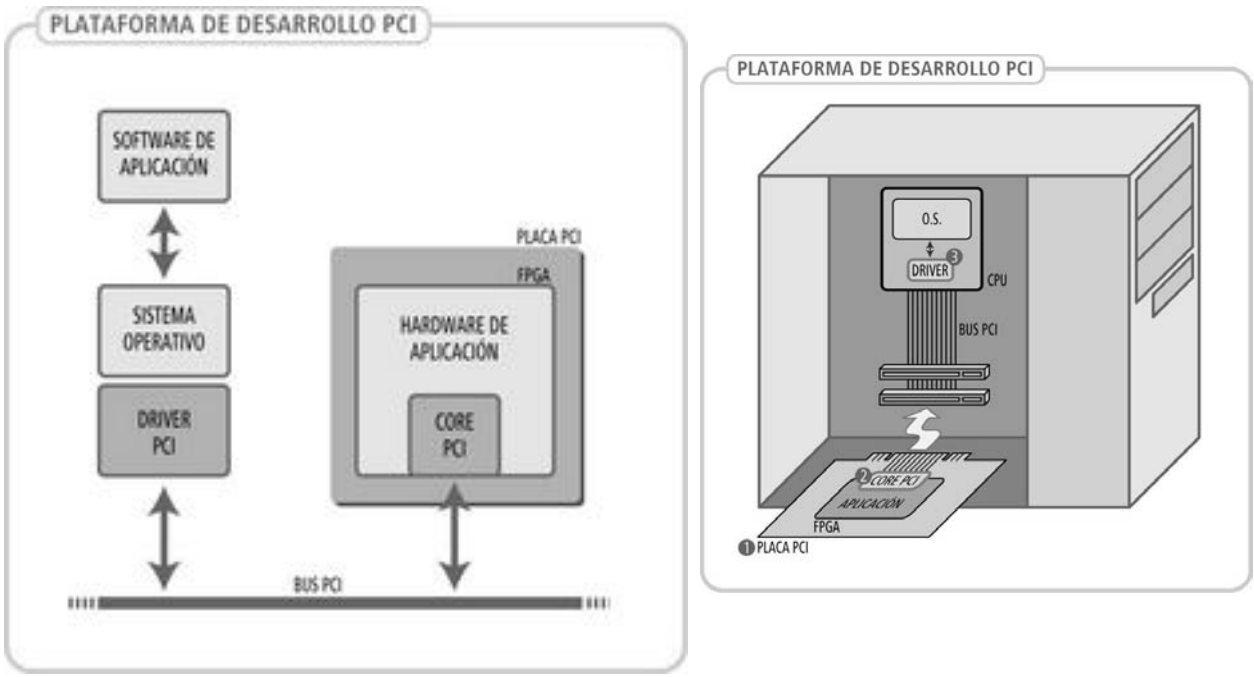


Figura 1: Esquema gráfico de los objetivos a llevar a cabo.

Una vez selecto el lenguaje se describen la estructura y metodología de trabajo de nuestro proyecto.

Los pasos que se siguieron para la implementación del trabajo fueron:

1- Aprendizaje del lenguaje VHDL.
2- Análisis de factibilidad del proyecto, investigando en la documentación de proyectos similares.
3- Estudio de la arquitectura del bus PCI y estudio de la norma.
4- Elección del kit apropiado en precio/prestación sobre el que basamos nuestro trabajo.
5- Diseño y codificación del CORE.
6- Simulaciones.
7- Debug y testing.
8- Mediciones y validación.

4.2-Selección de Herramienta:

Raggedstone1
Spartan-3 FPGA Development Board

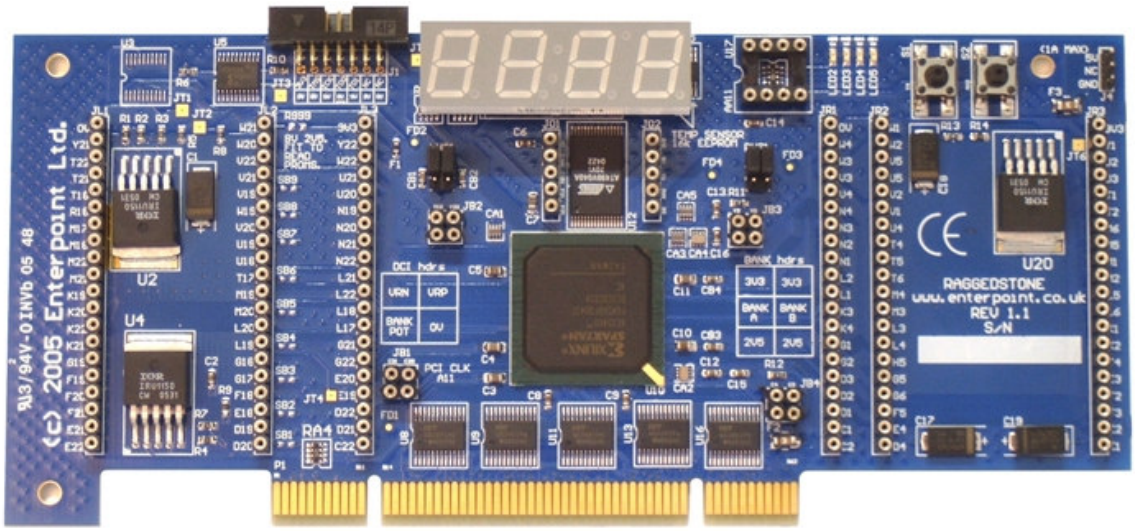


Figura 2: Raggedstone1 Spartan-3 FPGA Development Board.

Luego de un profundo análisis de mercado, se seleccionó el kit que muestra la figura 2, el mismo estaba integrado con una fpga XILINX XC3s400, de considerable tamaño, cabe mencionar que como se verá más adelante nuestro CORE completamente ocupa aproximadamente un 40 % de las capacidades del dispositivo, además funciona en ambos buses como puede observarse, se puede instalar en buses PCI de 3,3 v así como también en buses de 5v.

Dentro de sus principales características se destacan:

- FLASH memory, 4 Mbit.
- 1 x 16KBit serial EEprom fitted.
- 1 x LM75 Sensor de temperatura para múltiples aplicaciones incorporado.
- 4 x 7 SEGMENT digit LED – Con la posibilidad de removerlos para tener extras I/O.
- 4 LEDS de múltiple aplicación.
- Socket para utilización de modulo de Clock externo, o I/O adicionales.
- 2 Pushes switch para múltiple aplicación.
- Bancos de entrada salida configurables para 3,3v/2,5v
- Aproximadamente 120 I/O disponibles vía pin out + Display removable I/O.

4.3-Implementación:

La codificación y estructura del core están fundados en la norma PCI[3].

A continuación se muestra en la Figura 3 la FSM (Finite State Machine) propuesta por la norma:

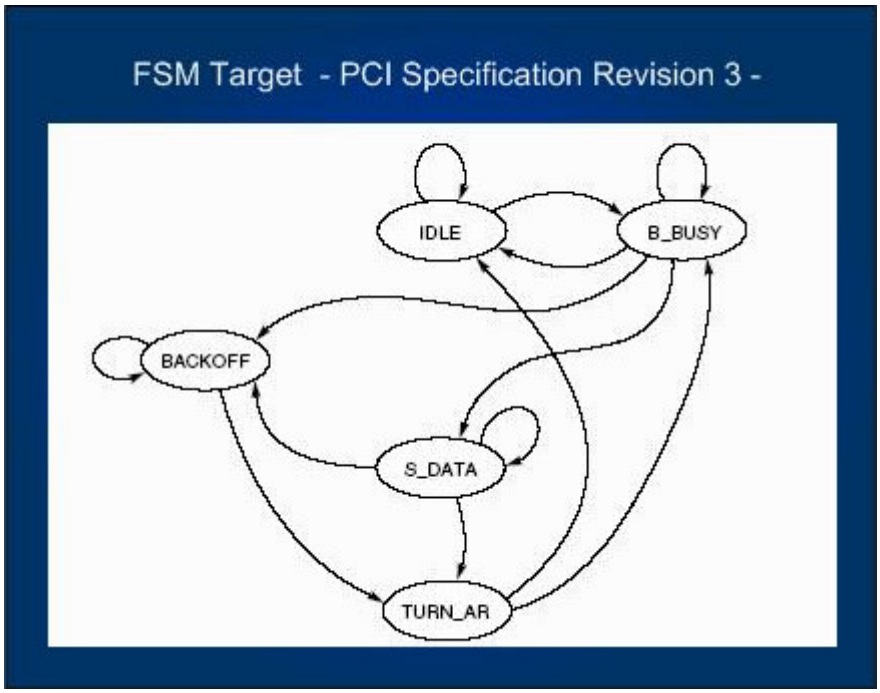


Figura 3: FSM sugerida por PCI Specification Revision 3.

Basándonos en esta estructura, nuestro CORE cuenta con una máquina de estados de desarrollo propio. Fue necesario sumar a la FSM un conjunto de Bloques Lógicos para cumplir las funcionalidades exigidas por la norma.

[3] PCI SPECIFICATION REVISIÓN 3 - PCI-SIG - www.pcisig.com/specifications

Las principales funcionalidades se observan en el siguiente diagrama (figura 4):

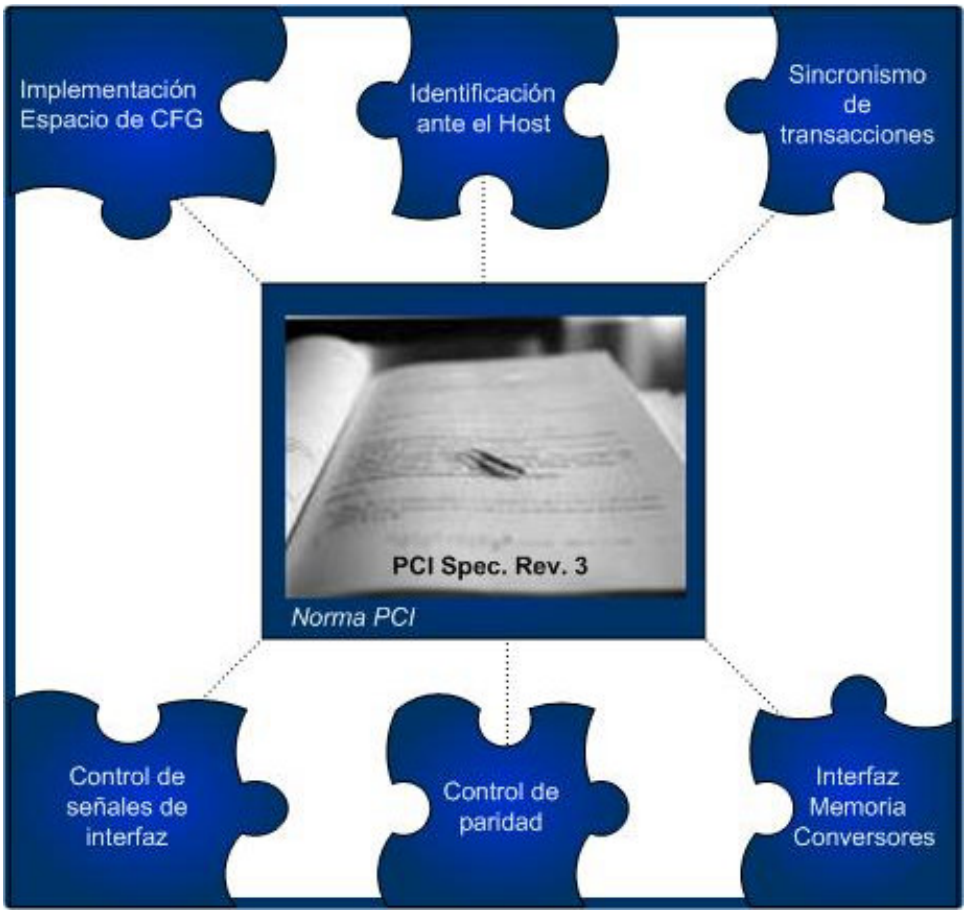


Figura 4: Diagrama funcional de Core PCI.

4.4-Arquitectura:

Para cumplir con nuestro diagrama funcional, implementamos nueve Bloques Lógicos con una estructura FLAT, en la cual en cada Bloque se desarrolla una tarea determinada en forma paralela y todos los Bloques son instanciados por un Bloque esquemático.
En la Figura 5 se puede apreciar la arquitectura implementada:

BLOQUE	**FUNCIÓN**
PCI	Esquemático – Instancia componentes
Parcontrol	Control de errores de paridad
Regdir	Registro de la dirección del dispositivo objeto de la transacción
C	Comparar direcciones en fase de direccionamiento
RdDatos	Manejo bidireccional de transferencia de datos
Fsm	Maquina de estados – Finite State Machine
Def	Definiciones
ConfSpace	Registro de espacio de configuración del dispositivo
Cout	Generar las combinaciones de señales de salida

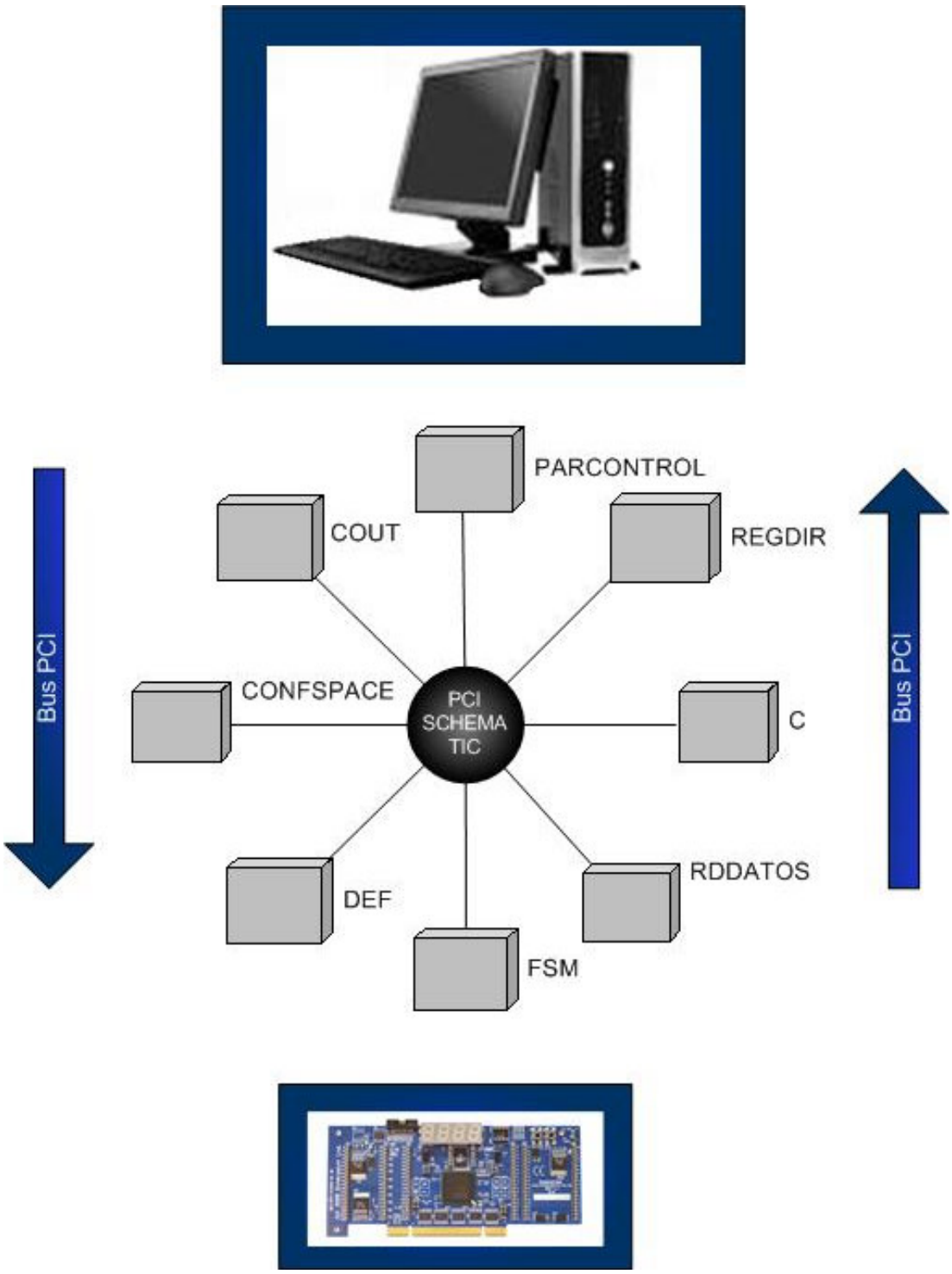


Figura 5: Estructura de bloques de Nuestro CORE.

4.5-Driver:

Para testear el funcionamiento bajo entorno windows se utilizo WinDriver[4] para desarrollar el Driver de nuestra tarjeta. Esta herramienta nos permitía leer los registros de configuración, y también nos da la posibilidad de leer y escribir los Base Address Registers (Espacio de configuración de nuestro dispositivo).

En la figura 6 se puede observar la información del espacio de configuración de nuestro dispositivo.

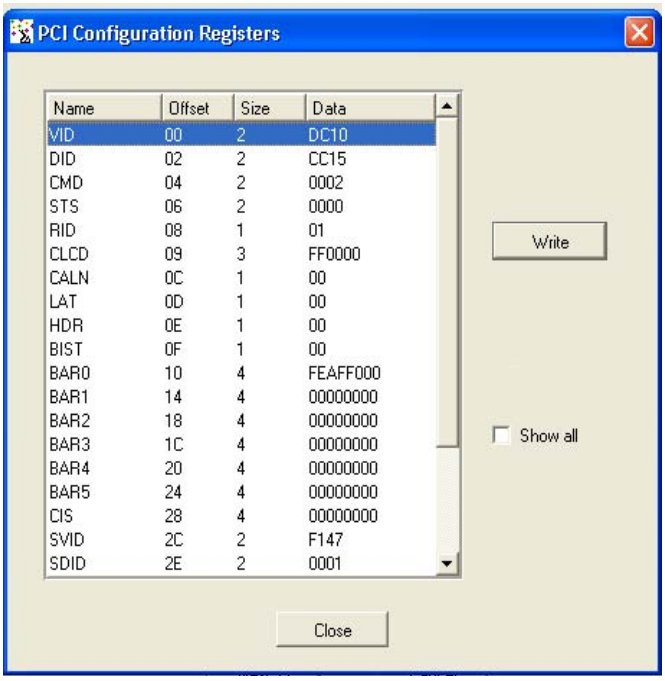


Figura 6: Lista de Registros de Configuración accedidos por WinDriver.

4.6-Aplicaciones:

Para validar el funcionamiento de nuestro CORE bajo el sistema operativo fue necesario implementar un nuevo driver y una serie de aplicaciones gráficas entre ellas Generador de Funciones – Osciloscopio Analizador de señal digitalizada. APIs desarrolladas (Figuras 7, 8 y 9).

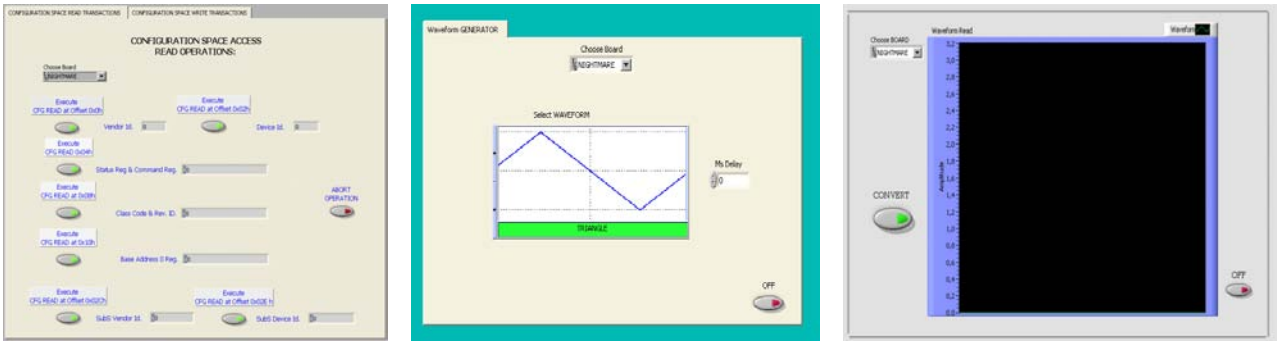


Figura 7, 8 y 9: Aplicaciones Gráficas.

[4] Driver Development Tools – Windriver - www.jungo.com

5-MEDICIONES

Para medir el tiempo de respuesta y el sincronismo de un dispositivo estándar PCI (Placa de video) se utilizo un ANALIZADOR LÓGICO, modelo LA4000/4280 80 channels, este equipo en particular constaba de 80 canales, adaptador a puerto USB y además un rate máximo de sampleo de 200Ms/s. Una vez observado el comportamiento se trabajó sobre nuestra tarjeta para igualar la respuesta temporal analizada.

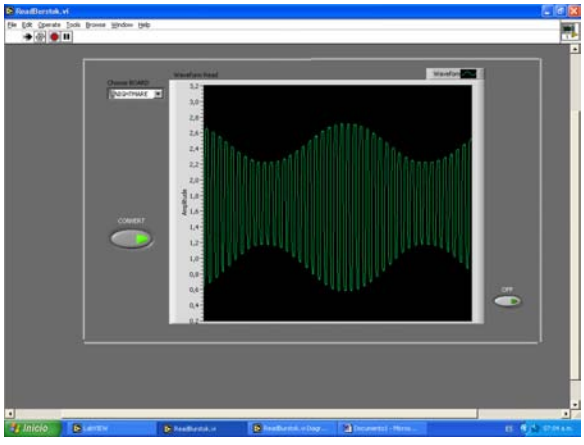
El resultado obtenido muestra que nuestro dispositivo se configura automáticamente en tiempo de arranque a una frecuencia de bus de 33 MHZ, cumpliendo con el grado de estándar, bajo el nivel mas rápido de respuesta temporal al maestro: Devsel Timing FAST[5].

Otra medición realizada en tiempo de compilación fue el espacio de pastilla ocupado por el CORE implementado, así como también el espacio total utilizado por el CORE más el código necesario para el desarrollo de las aplicaciones de almacenamiento de datos (buffer).

Los resultados obtenidos fueron:

El CORE sin aplicaciones empleaba aproximadamente el 20% de los recursos, mientras que el espacio total utilizado llegaba al 45% aproximadamente sumando a nuestro CORE las aplicaciones mencionadas en el párrafo anterior.

Para validar el funcionamiento del proyecto se implementaron dos placas conversoras una AD con el micro ADC Max1449 y otra DA con el micro DAC0800. Equipados conjuntamente con un par de generadores de funciones y un osciloscopio se corroboró el funcionamiento, digitalizando señales moduladas en amplitud de una frecuencia aproximada de 100 Khz, estando limitados no solo por las placas sino por el tiempo de muestreo del software y la mutiplexación del bus PCI, imágenes de este proceso se observan en las figuras 10 y 11.



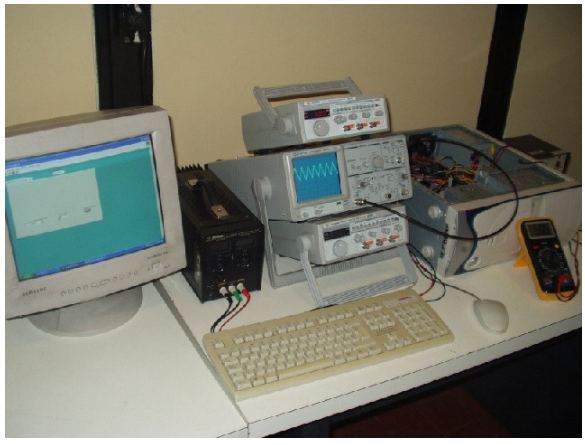
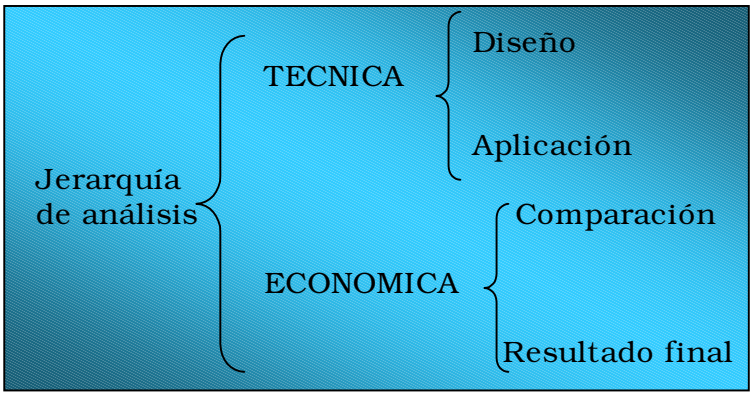


Figura 10 y 11: Transferencia bidireccional de datos – Análisis de señales y Generación de patrones de onda.

[5] Devsel Timing - PCI SPECIFICATION REV 3.0.

6-CONCLUSIONES:

Las conclusiones del trabajo pueden analizarse jerárquicamente de acuerdo a la siguiente estructura:



En el análisis técnico del diseño del proyecto, se destaca la selección de la FPGA para la implementación:

El criterio utilizado para la selección fue adecuado, seleccionándose una FPGA de nivel medio Spartan-3 XC3S400[6] (8.064 logic cells) y con un speed grade –4.

Se concluye en una selección correcta debido a que se superó las exigencias previstas, observando que el espacio en la pastilla en ningún caso supero el 50 %, lo que permite añadir futuras aplicaciones tipo add-on al core realizado, destacando que la versión inferior de FPGA limita seriamente el diseño, y con respecto a la velocidad de respuesta se mostró que con un speed grade -4 la pastilla responde correctamente al funcionamiento del bus de 33 Mhz superando los objetivos temporales determinados.

Profundizando el análisis técnico, debe destacarse que nuestro CORE cumplió los objetivos predeterminados de:

- Sincronismo con bus PCI
- Transferencia bidireccional de datos
- Compatibilidad con dispositivos PCI

Con las herramientas mencionadas puede desarrollarse el CORE PCI y el DRIVER correspondiente, y debe destacarse que con estos dos indispensables elementos: CORE - DRIVER pueden implementarse numerosas aplicaciones en el campo de la tecnología y las señales digitales.

Analizando económicamente el proyecto, se observa que el valor de lo obtenido en función al precio pagado por el mismo, costo-beneficio fue positivo, porque debe considerarse que el proyecto se llevo a cabo con la tarjeta PCI compatible más barata disponible en el mercado (u$d 90.-) y los resultados obtenidos superaron ampliamente todos los objetivos planteados en el proyecto, cumpliendo todos y cada uno de los requerimientos.

[6] Xilinx Spartan 3 Family
ww.xilinx.com/products/silicon_solutions/fpgas/spartan_series/spartan3_fpgas/index.htm

MINIX4RT: Real-Time Semaphores

Pablo A. Pessolani
Departamento de Sistemas de Información
Facultad Regional Santa Fe - Universidad Tecnológica Nacional
ppessolani@hotmail.com

Abstract

MINIX4RT is an extension of the well-known MINIX Operating System that adds Hard Real-Time services in a new microkernel but keeping backward compatibility with standard MINIX versions.
Semaphores are the primitive synchronization and mutual exclusion mechanism in many operating systems, but MINIX does not supply those facilities. Semaphores were added to MINIX4RT, and because it is a Real-Time Operating System, they must meet some processing requirements as dequeueing of waiting processes in priority order and avoiding the Priority Inversion problem. This article describes the Real-Time Semaphores facilities available on MINIX4RT, its design, implementation, performance tests and their results.

Keywords: Real-Time, Semaphores, Priority Inheritance, Priority Inversion.

Resumen

MINIX4RT es una extensión del conocido Sistema Operativo MINIX que incorpora servicios de Tiempo Real Estricto en un nuevo microkernel pero manteniendo compatibilidad con las versiones anteriores del MINIX estándar.
Los semáforos son el mecanismo primitivo para la sincronización y exclusion mutua en varios sistemas operativos, pero MINIX no brinda esa facilidad. Se adicionaron semáforos a MINIX4RT y, como éste es un Sistema Operativo de Tiempo Real, deben reunir ciertos requisitos de procesamiento tal como priorizar la remoción de procesos de las colas de espera y evitar el problema de Inversion de Prioridades. El presente artículo describe las facilidades de Semáforos de Tiempo-Real disponibles en MINIX4RT, su diseño, implementación, tests de desempeño y sus resultados.

Keywords: Tiempo Real, Semáforos, Herencia de Prioridades, Inversión de Prioridades.

1 INTRODUCTION

Real-Time Operating System (RTOS) services must consume a limited and guaranteed amounts of time. That deterministic timing behavior is the key difference against time sharing Operating Systems (OS).

MINIX4RT (previously named RT-MINIXv2) [1, 2] is a Real-Time (RT) version of the well known MINIX 2.0.2 [3] Operating System designed to teach concepts on RT-programming, in particular, those related to RT-kernels; but it can be usable as a serious system on resource-limited computers. It is a good tool to experiment with novel OS policies and mechanisms, and to evaluate the impact of architectural innovations.

The design constrains for MINIX4RT were:

- *Compatibility with MINIX*: All processes that run on MINIX must run on MINIX4RT without modifications and sensible performance impact.
- *Minimal MINIX source code changes*: As MINIX is often used in OS design courses, students have deep knowledge of its source code. Reducing the source code changes keep the student's experience to learn a MINIX based RTOS. Most new code must be added in separated functions with few changes in the original MINIX code. This constrain also helps for easier system updates for newer MINIX versions.
- *Source Code readability*: As MINIX4RT is focused for academic uses, its source code must be easily understood, perhaps sacrificing performance.

MINIX uses message passing as its central paradigm because it has a Client/Server microkernel based architecture.The messages have fixed sizes and strict copy to value semantics. In OS without Virtual Memory as MINIX, a message transfer implies a copy of the message from the sender's process address space to the destination's process address space. Because the copy is a time-consuming operation, it reduces the performance of simple synchronization or mutual exclusion primitives. Semaphores have a lower performance cost because they do not need that copy. Furthermore, as every operation in a hard RTOS, MINIX4RT RT-Semaphore primitives need to have deterministic execution and blocking times.

The rest of this work is organized as follows. Section 2 introduces on MINIX4RT. Section 3 presents background information about synchronization and mutual exclusion primitives on MINIX. Section 4 presents the proposed RT-Semaphore model. Section 5 is about RT-Semaphores basic data structures. Section 6 describes kernel primitives to operate on RT-Semaphores. Performance evaluation is presented in Section 7. Finally, Section 8 presents conclusions and future works.

2 OVERVIEW OF MINIX4RT

MINIX4RT provides the capability of running Real-Time and Non Real-Time (NRT) processes on the same machine [1]. The RT-processes are executed when necessary no matter what MINIX is doing.

The RT-microkernel works by treating the MINIX OS kernel as a task been executed under a small RTOS based on software emulation of interrupt control hardware. In fact, MINIX is like the *idle* process for the RT-microkernel been executed only when there are not any RT-processes to run. When MINIX requests the hardware to disable interrupts, the RT-microkernel intercepts that

request, records it, and returns to MINIX. If one of those “disabled” interrupts occurs, the RT-microkernel records its occurrence and returns without executing the MINIX interrupt handler. Later, when MINIX requests the hardware to enable interrupts, the RT-microkernel intercepts that request and executes all previously "disabled" handlers with recorded interrupts. This emulation avoid that MINIX can disable RT-interrupts imposing long latencies to the execution of RT-interrupt service routines and RT-processes.

The major features of MINIX4RT are summarized as follows:

Layered Architecture: MINIX4RT has a layered architecture that helps to change a component without affecting the others [1].

Real-Time Sub-kernel: A RT-microkernel that deals with interrupts, Interprocess Communications (IPC), time management and scheduling is installed below MINIX kernel. The advantage of using a microkernel for RTOS is that the preemptability is better, the size of the kernel becomes much smaller, and the addition/removal of services is easier [1].

Timer/Event Driven Interrupt Management: Device Driver writers can choice among two strategies of RT-Interrupt management [1].

Fixed Priority Hardware Interrupt Processing: A priority can be assigned to each hardware interrupt that let service then in priority order [1].

Two Stages Interrupt Handling: Interrupt can be serviced in two stages. The hardware interrupt handler (inside interrupt time) performs the first part of the needed work and a software Interrupt handler (outside interrupt time) does the remaining work [1].

Fixed Priority Real-Time Scheduling: Each process has an assigned priority. The RT-kernel schedules them in priority order with preemption [2].

Periodic and Non-Periodic RT-processing: A period can be specified for a periodic process; the RT-microkernel schedules it on period expiration [2].

Process and Interrupt Handlers Deadline Expiration Watchdogs: The use of watchdog processes is a common use strategy to deal with malfunctioning RT-processes. When a process does not perform its regular function in a specified time (*deadline*) another process (*watchdog*) is signaled to take corrective actions [2].

Timer Resolution Management Detached from MINIX Timer: A Timer interrupt of 50 Hz is emulated for the MINIX kernel even though the hardware Timer interrupt has a higher frequency [4].

Software Timers: There are system facilities named Virtual Timers (VT) used for time-related purposes as alarms, timeouts, periodic processing, etc. One particular feature of MINIX4RT is that it handles software timer actions in priority order [4].

Real-Time Interprocess Communications: MINIX4RT IPC uses unidirectional communication channels called Message Queues that handle messages in priority order and guarantee message delivery in a timely fashion and avoid the Priority Inversion problem [5].

Statistics and Real-Time Metrics: There are several facilities to gather information about the system status and performance.

Only NRT-process can be created and terminated under MINIX4RT. The RT-kernel does not add new System Calls to create RT-processes. On the other hand, a NRT-process is converted into a RT-process using the *mrt_set2rt()* System Call. Therefore a RT-process is managed by the RT-kernel and blocked for the MINIX kernel and, a NRT-process is managed by the MINIX kernel and

blocked for the RT-kernel. Before converting a process, several parameters (as priority, period, watchdog process, etc.) must be passed to the RT-kernel using the *mrt_setpattr()* System Call.

3 SYNCHRONIZATION AND MUTUAL EXCLUSION ON MINIX

Rendesvous Message Transfer is the basic mechanism that MINIX uses to communicate, synchronize and make mutual exclusion among Tasks, Servers and Users' processes, and to notify hardware interrupt occurrence.

Those primitives are implemented as the following kernel functions[3]:

- *mini_send(caller, destination, msg):* If the *destination* process is blocked waiting for that message from the caller, the message is copied from the *caller's* message buffer pointed by *msg* to the *destination's* message buffer, otherwise the *caller* process is blocked.

- *mini_rec(caller, sender, msg):* If the *sender* process is blocked trying to send a message to the *caller* process, the message is copied from the sender's buffer to the buffer pointed by *msg* and the *sender* process is unblocked, otherwise the *caller* process is blocked.

4 MINIX4RT SEMAPHORE MODEL

A semaphore is a kernel object that one or more processes can acquire or release for the purposes of synchronization or mutual exclusion. They constitutes the classic method for restricting access to shared resources in a multiprogramming environment. In a RT-environment semaphore operations need to have deterministic execution and blocking times.

MINIX4RT RT-Semaphores are implemented inside the RT-microkernel and do not use any MINIX IPC primitives because:

- *mini_send()* and *mini_rec()* kernel functions could change the caller's RT-process to a *READY* state for the MINIX kernel, therefore would be selected to execute by its NRT-scheduler ignoring all its RT-execution attributes.

- If a RT-process makes a request to a NRT-process using *mini_send()*, the RT-process must wait for the reply from the NRT-process running at NRT-priority. This behavior could produce an Unbounded Priority Inversion (explained in Section 6).

In the same manner, RT-processes are inhibited of making any MINIX System Calls (except *exit()*) because the use MINIX IPC primitives. For this reason, MINIX4RT offers two sets of facilities:

System Calls: To be used by NRT-processes to set the RT-environment or to get RT-statistics. These System Calls use MINIX primitives and does not have timing constraints.

Kernel Calls: To be used by RT-processes to provide RT-services. These Kernel Calls does not use MINIX primitives and does have timing constraints.

MINIX4RT Semaphores have the following features:

- Configurable dequeueing policy (Priority order or FIFO order).

- Basic Priority Inheritance Protocol (BPIP) support to avoid Unbounded Priority Inversion [6].

- Statistical counters of *up*s (also known as *signal*) and *down*s (also known as *wait*) operations on the semaphore.
- Timeout support.

To eliminate the allocation delay, the RT-kernel reserves a memory space (called the System Semaphore Pool) where semaphore objects are stored.

5 RT-SEMAPHORE DATA STRUCTURES

MINIX4RT defines new data structures to operate with RT-Semaphores. It defines RT-kernel data structures and Userspace data structures as are described in the following sections.

5.1 RT-Semaphore Kernel Data Structure

The RT-microkernel defines a RT-Semaphore Descriptor data structure that have the following fields and data type definition:

```
struct MRT_sem_s {
        int             index;          /* semaphore ID                                 */
        int             value;          /* semaphore Value                              */
        priority_t      priority;       /* Ceiling priority - for future uses           */
        unsigned int    flags;          /* semaphore policy flags                       */
        int             owner;          /* semaphore owner                              */
        long            ups;            /*  #  of sem up() calls                        */
        long            downs;          /*  #  of sem down() calls                      */
        MRT_proc_t      *carrier;       /* the process that locked mutex semaphore      */
        link_t          alloclk;        /* Allocated list link                          */
        link_t          locklk;         /* Locked list link                             */
        char            name[MAXPNAME];     /* name of the semaphore                    */
        plist_t         plist;          /* Priority List of waiting process             */
        };
typedef struct MRT_sem_s MRT_sem_t;
```

- *index:* Identifies the Semaphore Descriptor into the System Semaphore Pool.
- value: The semaphore value that can be set by the *mrt_semalloc()* System Call. It is increased by one for each *mrt_semup()* System Call or *MRT_semup()* RT-Kernel Call. It is decreased by one for each *mrt_semdown()* System Call or *MRT_semdown()* RT-Kernel Call.
- *priority*: The ceiling priority used by the Priority Ceiling Protocol and the Semaphore Inheritance Protocol not implemented in the current version.
- *flags*: RT-Semaphore policy flags. It is an OR of the following bits:
  - *SEM_PRTYORDER*: If it is set the waiting RT-processes will be woken up in priority order, otherwise they will be woken up in First Come First Served (FCFS) order.
  - *SEM_MUTEX*: If it is set the RT-Semaphore will be used as a *mutex*, otherwise it will be a counting RT-semaphore.

- *SEM_PRTYINHERIT*: If it is set the RT-kernel applies the Basic Priority Inheritance Protocol to RT-Semaphore operations. This option is valid only if the *SEM_PRTYORDER* and the *SEM_MUTEX* bits are set.

- *owner*: The process which makes the *mrt_semalloc()* System Call.
- *ups* and *downs*: Statistical counters of *MRT_semup()* and *MRT_semdown()* RT-kernel calls since the RT-Semaphore allocation.
- *carrier*: The process that has locked the *mutex* RT-Semaphore.
- *alloclk*: A data structure to build a linked list of allocated RT-Semaphores. It is also used to insert/remove a RT-Semaphore into/from the Free list of the System Semaphore Pool.
- *locklk*: A data structure to build a linked list of RT-Semaphores locked by a RT-process.
- *name*: A name assigned to a RT-Semaphore.
- *plist*: A data structure to build a priority list of waiting RT-processes.

5.2 RT-Semaphore Userspace Data Structure

MINIX4RT defines several Userspace Data Structures to operate on RT-Semaphores as is described in the following sections.

5.2.1 RT-Semaphore Attributtes Data Structure

The fields of RT-Semaphore Attributtes data structure have the same meanings of the RT-Semaphore Descriptor data structure. It is used by the *mrt_semalloc()* and the *mrt_semattr()* system calls.

```
struct mrt_semattr_s {
        int             value;          /* semaphore Value                  */
        unsigned int    flags;          /* semaphore policy/status flags    */
        priority_t      priority;       /* Ceiling priority - for future uses */
        char            name[MAXPNAME];     /* name of the semaphore        */
                };
typedef struct mrt_semattr_s mrt_semattr_t;
```

5.2.2 RT-Semaphore Statistics Data Structure

This data structure is used to get RT-Semaphore statistics. It is used by the *mrt_semstat()* system call.

```
struct mrt_semstat_s {
        long            ups;            /* total #  of sem up() calls       */
        long            downs;          /* total #  of sem down() calls     */
        int             maxinQ;         /* maximun # of process enqueued    */
                };
typedef struct mrt_semstat_s mrt_semstat_t;
```

- *ups* and *downs*: Statistical counters of *mrt_semup()* and *mrt_semdown()* system calls since the RT-Semaphore allocation.

- *maxinQ:* The maximun number of waiting RT-processes enqueued into the RT-Semaphore list.

5.2.3 RT-Semaphore Internal Data Structure

This data structure is used to get the internal status of a RT-Semaphore. It is used by the *mrt_semint()* system call.

```
struct mrt_semint_s {
        int             index;      /* semaphore ID                */
        int             owner;      /* semaphore owner             */
        int             inQ;        /* # of process enqueued       */
                };
typedef struct mrt_semint_s mrt_semint_t;
```

- *index:* Identifies the Semaphore Descriptor into the System Semaphore Pool.
- *owner*: The process which makes the *mrt_semalloc()* System Call.

5.2.4 RT-Semaphore Down Data Structure

This data structure is used by the *mrt_semdown()* Kernel Call.

```
struct mrt_down_s {
        int             index;      /* semaphore ID                */
        lcounter_t      timeout;    /* timeout in ticks            */
                };
typedef struct mrt_down_s mrt_down_t;
```

- *index:* The identification of the RT-Semaphore.
- *timeout:* A timeout in Timer ticks can be specified to wait for the request RT-Semaphore.

5.3 RT-Semaphore Waiting RT-Processes Priority List

To manage the waiting RT-Processes on a RT-Semaphore, the RT-kernel uses a Priority List Data Structure (see Figure 1):

On insertion operations, the *priority*-th bit in the bitmap is set and the Process Descriptor is appended to the Priority List in accordance with its *priority* field.

Finding the highest priority RT-process into the priority list is therefore only a matter of finding the more most significant bit set into the bitmap. Because the number of priorities is fixed, the time to complete a search is constant and unaffected by the number of RT-processes into the Priority List.

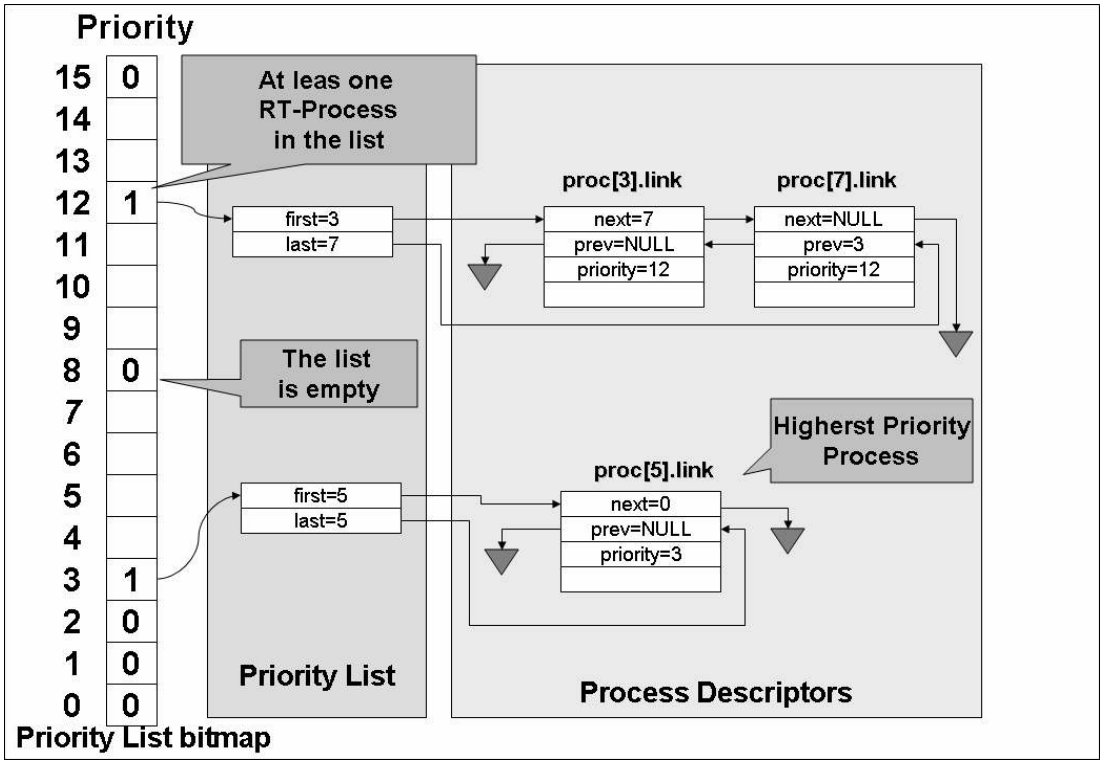


Figure 1: RT-Semaphore Waiting RT-Processes Priority List

6 RT-SEMAPHORE KERNEL CALLS

In many RT-applications, there are resources that must be shared among processes in a way that prevents more than one process from using the resource at the same time (mutual exclusion).

The Unbounded Priority Inversion problem is an unwanted situation where a higher priority process waits for a semaphore locked by a lower priority process and a medium priority process preempt it delaying the semaphore release and therefore the high priority process execution.

There has been developed many mechanisms to avoid it. Sha, Rajkumar and Lehosky [7] suggest two protocols to avoid the Unbounded Priority Inversion problem. The are the Basic Priority Inheritance Protocol (BPIP) and the Priority Ceiling Protocol (PCP).

Under the BPIP, if a lower priority process blocks a higher priority process, the lower priority process inherits the priority of the higher priority process for the duration of its critical section. The BPIP potentially requires priorities to be modified when processes try to lock a locked semaphore. The process that have locked the requested semaphore may inherit the higher priority among the petitioner's priorities. To achieve the correct behavior and be compliance with BPIP, priority inheritance needs to be a transitive operation. Therefore, the RT-kernel must search across the chain of petitioner processes to apply the priority inheritance until it finds the process that has no pending requests. MINIX4RT provides RT-Semaphore primitives that are compliance with the BPIP offering a deterministic timing behavior.

6.1 mrt_semdown() Kernel Call

The *mrt_semdown()* Kernel Call decrease the semaphore's count by one. If the resulting semaphore value drops below zero, the caller process will block and its descriptor is inserted into the RT-Semaphore Waiting RT-Processes Priority List.

For RT-Semaphores used as mutexes, the process that has locked the RT-semaphore increase its priority to the caller's priority, if it is higher than its owns. If that RT-process is blocked waiting for another RT-semaphore, the Priority Inheritance Protocol is applied to all RT-process in the chain.

A timeout in Timer ticks can be specified to wait for the RT-Semaphore release. A special value of *MRT_NOWAIT* can be specified to return without waiting if the RT-semaphore is locked by other RT-process. To wait until the RT-Semaphore release, *MRT_FOREVER* must be specified as *timeout*. On timeout expiration:

The RT-process descriptor is removed from RT-Semaphore Waiting RT-Processes Priority List.

The caller is unblocked returning and *E_MRT_TIMEOUT* error code.

For RT-Semaphores used as mutexes, the priority of the RT-process that had locked the RT-Semaphore is set to the highest priority waiting process into RT-Semaphore Waiting RT-Processes Priority List or its base priority specified in the *MRT_setpattr()* System Call.

6.2 mrt_semup() Kernel Call

If the semaphore value is lower than zero, its absolute value indicates the number of waiting RT-process blocked trying to *down* the semaphore. The *mrt_semup()* Kernel Call increases the semaphore's count by one, removes the highest priority process (if the *SEM_PRTYORDER* bit is set in *flags*) or the first process into RT-Semaphore Waiting RT-Processes Priority List and unblocks it.

For RT-Semaphores used as mutexes, if the BPIP had raised the caller's priority when it locked the semaphore, its priority is returned to it base priority specified in the *MRT_setpattr()* System Call.

7 PERFORMANCE EVALUATION

This section describes the tests performed on MINIX4RT Semaphores and their results. The RT-Semaphore operations performance was tested with four kinds of system setups/policies (see Table 1), with and without timeout settings, with and without applying BPIP. The tests consist in 10000 rounds of the Producer/Consumer algorithm (two *down* operations and two *ups* operations per process per round).

Table 1: Setups and Policies of Semaphore Operations Performance Tests

Test Name	With Timeout	Priority List/FIFO	Priority Inheritance
TEST1	No	Priority List	No BPIP
TEST2	Yes	Priority List	No BPIP
TEST3	No	Priority List	BPIP
TEST4	Yes	Priority List	BPIP

The tests were performed under different kinds of loads on the tested system (see Figure 2):

1. *Without Load (NOLoad):* All unneeded processes are killed before the test.

2. *CPU Load(CPULoad)*: A NRT-script loads the CPU without any I/O operation.
3. *I/O Disk Load(HDLoad)*: A NRT-process access files on the hard disk.
4. *I/O RS232e Load (RSLoad)*: A NRT-file transfer over the serial port at 19200 Kbps.

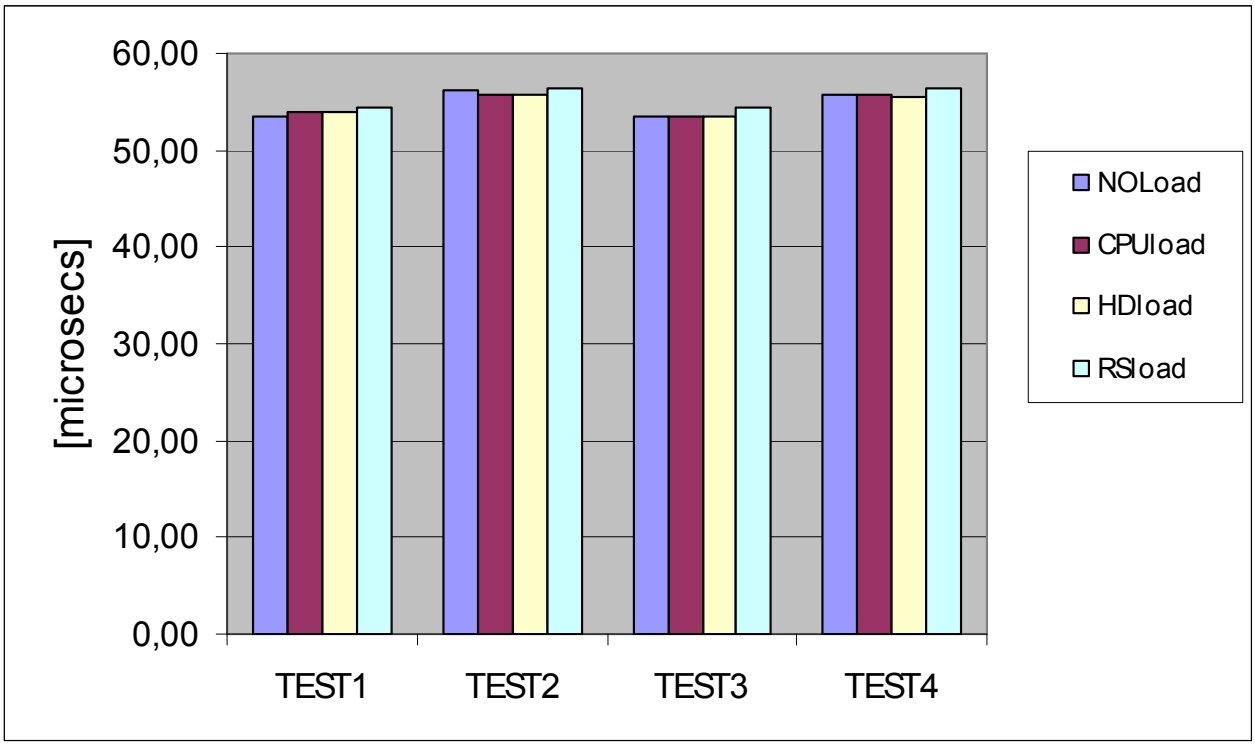


Figure 2: Down-Up pair processing time.

Table 2 presents Down-Up pair processing processing times.

Table 2: Down-Up Pair Processing Times [µs]

	TEST1	TEST2	TEST3	TEST4
NOLoad	53,42	56,25	53,52	55,75
CPUload	54,02	55,67	53,57	55,67
HDload	54,05	55,65	53,57	55,57
RSload	54,45	56,42	54,32	56,45

All the tests were carried on with the Programmable Interval Timer was set up at 1000[Hz]. This fact implies the execution of the Timer Interrupt Service Routing 1000 times by second adding an significative overhead to the measurements, but presents a more realistic scenario. Other tests performed on MINIX4RT showed an average Timer Interrupt Service Time of 32[µs].

The equipment used for the tests was an IBM Model 370C Notebook, Intel® DX4 75 MHz, AT Bus, Memory 8 MB, and MINIX4RT (Kernel 12052007). In spite of the equipment is quite old , it allows performance comparisons against reports of other systems with similar hardware.

Sacha [8] reports QNX *signal* times about 40-45[µs] on a 486/66 MHz. His results show the same order of magnitud than the tests results on MINIX4RT considering that they include *down* time plus *up* time and the CPU clock difference.

8 CONCLUSIONS AND FUTURE WORKS

MINIX has proved to be a feasible testbed for OS development and extensions that could be easily added to it. In a similar way, MINIX4RT has an architecture that can be used as a starting point for adding RT-services. In spite of it was designed for an academic environment, it can be optimized for production systems even in embedded systems. MINIX4RT combines Hard Real-Time with the standard MINIX platform so time sensitive control algorithms can operate together with background processing without worrying about interference.

MINIX4RT algorithms were developed to minimize priority inversion to meet applications with strict timing constrains. A sample of that is the use of Priority Lists and the use of the Basic Priority Inheritance Protocol.

The RT-microkernel has basic features as Interrupt Management, Process Management, Time Management, RT-IPC and Statistics gathering making it a good choice to conduct coding experiences in Real-Time Operating Systems courses.

Near future works on MINIX4RT are:

- *Operating System Profiling:* Runtime profiling is a key technique to prove new concepts, debug problems, and optimize performance.

- *Port Real-time to MINIX3*: The current version of MINIX has a more strict compliance with a Client/Server microkernel based Opertating System. That changes cause the need of rewrite some components of MINIX4RT code to able to run under MINIX3.

REFERENCES

[1] Pessolani, Pablo A, "*RT-MINIXv2: Architecture and Interrupt Handling",* 5th Argentine Symposium on Computing Technology, 2004.

[2] Pessolani, Pablo A., "*RT-MINIXv2: Real-Time Process Management and Scheduling*", 6th Argentine Symposium on Computing Technology, 2005.

[3] Tanenbaum Andrew S., Woodhull Albert S., "*Sistemas Operativos: Diseño e Implementación*" 2da Edición, ISBN 9701701658, Editorial Prentice-Hall , 1999.

[4] Pessolani, Pablo A., "*MINIX4RT: Time Management and Timer Facilities*", 7th Argentine Symposium on Computing Technology, 2006.

[5] Pessolani, Pablo A., "*MINIX4RT: Real-Time Interprocess Communications Facilities*", Workshop de Arquitecturas, Redes y Sistemas Operativos, XII Congreso Argentino de Ciencias de la Computación, 2006.

[6] Mark W. Borger, Ragunathan Rajkumar. *"Implementing Priority Inheritance Algorithms in an Ada Runtime System*", Technical Remailbox . CMU/SEI-89-TR-15. ESD-TR-89-23. Software Engineering Institute Carnegie Mellon University, 1989.

[7] Sha, L., Lehoczky, J.P., and Rajkumar, R. "*Priority Inheritance Protocols: An Approach to Real-Time Synchronization*". Tech. Rept. CMU-CS-87-181, Carnegie Mellon University, Computer Science Department, 1987.

[8] Krzysztof M. Sacha, "*Measuring the Real-Time Operating System Performance*", Institute of Control and Computation Engineering, Warsaw University of Technology, Poland, 1995.

PROPOSTA DE PROTOCOLOS DE COMUNICAÇÃO PARA UM AMBIENTE DE COMPUTAÇÃO UBÍQUA PARA AUTOMAÇÃO RESIDENCIAL

Celso Roberto Perez
Derci de Oliveira Lima

Instituto Pernambucano de Pesquisa, Pós-Graduação
e Empreendedorismo – I3PE
Rua Ernesto de Paula Santos 960, 1º. Andar, sala 102,
Caixa Postal 34 - Recife – PE – Brasil
CEP 51021-330

celsope@gmail.com
derci_lima@terra.com.br

Resumo: *Um ambiente de computação ubíqua tem como pressuposto que a computação está em todos os equipamentos e dispositivos, e estes estão interagindo entre si e com as pessoas. A conexão entre os objetos e seres humanos se dá instantaneamente, quando estes entram no ambiente ubíquo. Um protocolo de comunicação para fazer a ligação entre os vários dispositivos, de vários fabricantes, neste ambiente, deverá ser a chave para que esta conectividade aconteça. Na automação industrial, o protocolo RS 485, é muito utilizado. Apesar de algumas semelhanças nos propósitos, a automação residencial tem conceitos diferentes ao da industrial. O protocolo I2C, da tecnologia com fio, desenvolvido pela Philips, surge como opção para a automação residencial. Protocolos de tecnologia sem fio, como bluetooth e zigbee, devido as suas característica de cobertura, apresentam-se como complemento para esta automação. Este artigo visa fornecer uma visão das características necessárias dos protocolos de comunicação para um ambiente de computação ubíqua na automação residencial. Nele será relatado os estudos, desenvolvimento e implantação de dispositivos de automação residencial que ocorre na QualiHouse Automação Residencial, empresa incubada na Trilha, pre-incubadora da Faculdade Integrada do Recife - FIR..*

Palavras-chave: *Computação Ubíqua, Protocolo de Comunicação, Automação, I2C.*

Abstract: *An ubiquotis computation environment has as estimated that the computation is in all the equipment and devices, and these are interacting between itself and with the people. The connection between objects and human beings if gives instantanly, when these enter in the ubicomp environment. A protocol of communication to make the linking between the some devices, of some manufacturers, in this environment, will have to be the key so that this conectivity happens. In the industrial automation, protocol RS 485, very is used. Although some similarities in the intentions, the residential automation has different concepts to the one of the industrial. Protocol I2C, of the technology with wire, developed for the Philips, appears as option for the residential automation. Protocols of technology without wire, as bluetooth and zigbee, had its characteristic of covering, are presented as complement for this automation. This article aims at to supply to a vision of the necessary characteristics of the communication protocols an ubicomp environment in the residential automation. In it will be told the studies, development and implantation of devices of residential automation that occurs in the QualiHouse Residential Automation, pre-incubated company in the Trilha, pre-incubator of the Faculdade Integrada do Recife – FIR.*

Keywords: *Ubiquiti Computing, Comunication Protocols, Automation, I2C.*

1. INTRODUÇÃO

Os ambientes onde comportamentos se alteram automaticamente, sem que os usuários precisem explicitar seus desejos, são chamados de ambientes inteligentes. Quando toda a interação com o ambiente acontece através dos comportamentos automáticos, caracteriza a computação invisível. Com a mobilidade física dos equipamentos, mobilidade lógica das aplicações somadas ao ambiente inteligente nos dá a computação ubíqua.

Mark Weiser em 1988 foi o primeiro autor a usar o termo computação ubíqua. Ele o usou para descrever sua idéia de tornar os computadores onipresentes e invisíveis. A idéia era a de não precisar existir visualmente um computador para que o usuário pudesse interagir, em outras palavras, trabalhar ou se divertir. Não bastava ter uma interface amigável, precisava ir além. Com mais aprimoramento do que o que existe na realidade virtual. Nada de teclado, mouse, monitor ou fios ligados ao usuário. "É fazer seu trabalho com o auxílio dos computadores sem se preocupar em trabalhar nos computadores" [1].

A automação residencial vem ganhando força nas duas últimas décadas, afirmando seus conceitos próprios, e não mais os da automação industrial. É considerada por muitos integradores, que são pessoas especializadas em colocar para funcionar de forma interligada tecnologias adversas, como uma tarefa difícil, pois em automação residencial cada cliente tem um contexto diferente, equipamentos diferentes, aplicações diferentes. É preciso analisar o que cada cliente deseja da automação e desenhar uma solução para aquela situação proposta. Haverá clientes priorizando a segurança de sua residência, com sensores e câmeras de vídeo espalhadas pelo quintal, e outros dando prioridade ao entretenimento, instalando telões e distribuindo som e vídeo nos cômodos da casa.

A comunicação dos diversos equipamentos e dispositivos dentro do ambiente de computação ubíqua é o problema a ser resolvido. Os fabricantes dos diversos equipamentos elétricos e eletrônicos que equipam uma residência, ainda não convergiram para uma tecnologia única, aonde os integradores da automação residencial pudessem utilizar desta facilidade.

A proposta deste projeto de pesquisa é a implementação dos conceitos e princípios da computação ubíqua para aplicação dentro de uma Smart House (casa inteligente). Para isto serão estudadas e implementadas as tecnologias dos diversos dispositivos dentro da casa. A empresa QualiHouse Automação Residencial, pré-incubada na Trilha da Faculdade Integrada do Recife, é a responsável pelo desenvolvimento do sistema de controle que será utilizado como modulo de testes e showroom das tecnologias envolvidas. Neste projeto, o trabalho do bolsista consiste em estudar, propor, desenvolver e implementar, escolhendo quais os protocolos de comunicação que interligariam com eficiência os diversos dispositivos dentro do ambiente ubíquo.

Para implementar e testar os protocolos aqui propostos, esta sendo desenvolvido um módulo em protoboard, que são mesas de montar circuitos eletrônicos experimentais, na empresa QualiHouse, chamado de ponto de controle. O ponto de controle utilizará o protocolo de comunicação I2C, interno para acessar a memória e o relógio de tempo real e externo quando estiver conectado aos pontos de automação. Para cobrir distancias maiores, poderão ser utilizados extensores da Philips para interligar o ponto de controle aos pontos de automação. Também utilizará o protocolo RS232 para comunicação com um computador.

Este documento está organizado da seguinte forma: na segunda seção é contextualizado um ambiente de computação ubíqua para automação residencial. A terceira seção trata sobre as características e necessidades da comunicação para a automação residencial no ambiente de

computação ubíqua. Na quarta seção o assunto são os protocolos com e sem fio, suas vantagens e desvantagens para esta aplicação. A quinta seção foi dedicada ao projeto que está sendo desenvolvido em parceria com a empresa pré-incubada na Trilha da Faculdade Integrada do Recife. Ali estão relatados os planos iniciais e as mudanças de plano. A sexta seção, como conclusão, faz uma reflexão sobre o tema, o resultado alcançado, problemas encontrados para realização deste projeto e como continuar as pesquisas deste projeto. Na sétima seção foram colocadas as referências bibliográficas. Finalmente na última seção estão colocados os mini currículos dos autores deste trabalho.

2. O AMBIENTE DE COMPUTAÇÃO UBÍQUA

Nesta seção será abordado o tema do ambiente ubíquo para a automação residencial. Serão introduzidos conceitos como computação pervasiva, mobilidade física e lógica, contexto de execução das aplicações e ambientes inteligentes para explicar os conceitos da computação ubíqua.

A computação pervasiva é caracterizada como aquela onde o ambiente computacional do usuário está disponível em qualquer lugar, a qualquer tempo. Neste, existe a associação dos dois tipos de mobilidade: a mobilidade física dos equipamentos e a mobilidade lógica dos dados e aplicações (programas). As aplicações são móveis, distribuídas, conscientes do contexto, e tomam a forma siga-me. Por exemplo, um usuário deste sistema poderia fazer com que o seu canal preferido de música o seguisse por qualquer cômodo do ambiente, sem que ele precisasse ligar e desligar qualquer equipamento de som, em seu deslocamento pelo ambiente. A adaptação é o requisito mais importante para que estas novas aplicações atendam seus usuários [2]. Esta mobilidade trouxe novos desafios para a produção de aplicações que deverão se adaptar a estes ambientes.

A mobilidade física está relacionada a localização. A localização atual do usuário, dentro do ambiente, determina qual o contexto de execução das aplicações que este usuário exigirá. Neste cenário, a noção de adaptação está relaciona com reação à troca de estado dos elementos que compõem o contexto da aplicação e que alteram seu comportamento.

Abowd & Mynatt [3] descrevem como obter informações contextuais analisando cinco aspectos primordiais, os quais são chamados de cinco Ws:

- Who (quem): deve prover informações contextuais de todas as pessoas envolvidas em uma dada atividade assistida por computador.
- What (o quê): tem como função identificar o que o usuário está fazendo.
- Where (onde): como mencionado, o contexto de localização é o mais utilizado pelos sistemas sensíveis a contexto.
- When (quando): o contexto temporal é importante para indexar uma informação capturada, por exemplo, informar por quanto tempo um usuário esteve em um determinado local;
- Why (por que): não é só perceber o que o usuário faz, mas principalmente o porquê ele faz aquela ação.

Nestes ambientes onde comportamentos automáticos acontecem, sem que os usuários precisem explicitar seus desejos, são chamados de ambientes inteligentes. Quando toda a interação com o ambiente acontece através dos comportamentos automáticos, caracteriza a computação invisível. Com a mobilidade física dos equipamentos, mobilidade lógica das aplicações somadas ao ambiente inteligente nos dá a computação ubíqua [1] [2] [4].

Mark Weiser em 1988 foi o primeiro autor a usar o termo computação ubíqua. Ele o usou para descrever sua idéia de tornar os computadores onipresentes e invisíveis. A idéia era a de não precisar existir visualmente um computador para que o usuário pudesse interagir, em outras palavras, trabalhar. Não bastava ter uma interface amigável, precisava ir além. Com mais aprimoramento do que o que existe na realidade virtual. Nada de teclado, mouse, monitor ou fios ligados ao usuário [1]

A tabela 1 apresenta as relações existentes entre os conceitos de computação aqui abordados em relação aos atributos de mobilidade e grau de inteligência ou funcionalidades dos dispositivos envolvidos em cada conceito.

Características	Computação Pervasiva	Computação Móvel	Computação Ubíqua
Mobilidade	Baixa	Alta	Alta
Grau de inteligência	Alta	Baixa	Alta

Tabela 1 - Relação entre os conceitos apresentados

3. A COMUNICAÇÃO EM AMBIENTE UBÍQUO

Para que todos os dispositivos e equipamentos dentro do ambiente ubíquo possam se comunicar, é necessário que existam protocolos de comunicação que todos os elementos deste contexto possam compreender. Nesta seção abordamos necessidades e características desta comunicação que influem na escolha dos protocolos.

A discussão sobre protocolos de comunicação entre os equipamentos e conectividade entre diversas tecnologias, têm propiciado que os grandes fabricantes de circuito integrado, eletrodomésticos e eletroeletrônicos, os provedores de acesso as redes, prestadores de serviço de software e telecomunicações, se associem em diferentes grupos de trabalho para criar padrões de comunicação comuns entre eles. Estes grupos de estudos criam as regras que devem ser seguidas para que seus equipamentos possam se comunicar estando ligados em redes. A área de automação residencial é a mais beneficiada por estes estudos [5].

As novas tecnologias propiciam conforto, praticidade, produtividade, economia, eficiência, rentabilidade com adicional valorização do empreendimento para seu usuário. Com a automação residencial o que se objetiva, segundo Bettoni [6], é a integração de tecnologias de acesso à informação e entretenimento, com otimização dos negócios, da Internet, da segurança, além de total integração da rede de dados, voz, imagem e multimídia. Isto só é conseguido através de um projeto único que envolva a infra-estrutura, dispositivos e software de controle.

4. PROTOCOLOS DE COMUNICAÇÃO

Nesta seção discorremos sobre os protocolos do meio físico, também chamados de protocolos de baixo nível. Primeiro apresentamos os protocolos de comunicação serial, da tecnologia com fio, que poderão ser utilizados para compor um ambiente ubíquo. Na segunda parte serão apresentados alguns protocolos de comunicação sem fios, também conhecidos por wireless, que além de dar mobilidade ao sistema de controle e monitoramento do ambiente, poderão fazer conexões de dispositivos que não possam ser alcançados pela rede cabeada.

4.1 Tecnologias com fio

Um protocolo muito conhecido e presente nos computadores antigos e novos é o RS 232 [9] [10]. É serial, ponto a ponto, de baixo custo, porém tem alcance curto (uns 15 metros no máximo) devido a suscetívidade a ruídos. Necessita de linhas separadas para transmissão e recepção. Nos computadores, ele trabalha com o chip UART que tem a função de converter os bits paralelos do computador para a saída serial do RS232 e vice versa. Em computadores mais antigos, o chip UART é o gargalo das taxas de transferência, pois trabalham com um cache pequeno.

O protocolo RS422 [10] também é serial, ponto a ponto. É considerado como um melhoramento do RS232, tendo linhas de transmissão e recepção balanceadas, ficando mais resistentes a ruídos. Consegue alcance de 1200 metros. É usado como extensor do RS232. O protocolo RS485 [10] é uma evolução do RS422, sendo serial, porém multi-ponto. Pode ter até 32 dispositivos em rede, trabalhando em barramento diferenciado de dois fios, transmissão separada da recepção, consegue alcance de 1200 metros. Os dispositivos escravos são acessados por endereçamento único.

O X-10 - PLC (Power Line Carrier) [11] é um protocolo que usa a linha de tensão da residência como meio físico para comunicação entre o transmissor e o receptor. Aceita até 256 dispositivos para endereçamento. Trabalha com uma freqüência mais alta que a da rede para enviar comandos. Tem evoluído nos últimos anos. Porém ainda atua com pouca flexibilidade, fazendo só abrir e fechar contatos.

O SPI [12] é um protocolo serial, multi-ponto, desenvolvido pela Motorola. Utiliza 2 fios comuns a todos os dispositivos, alem de um fio para endereçamento de cada dispositivo slave.

O protocolo IEEE 1394 [13] também conhecido como Firewire, da Apple, é uma porta plug and play, isto é, os dispositivos conectados a ela já possuem os drives de instalação para seu funcionamento. É também hot, que significa que os dispositivos podem ser conectados e desconectados sem precisar desligar o equipamento. Muito usada para transmissão de vídeos devido a sua grande largura de banda.

O protocolo I²C [13] [15] original da Phillips, esta licenciado para um grande número de fabricantes. Possui um barramento de dois fios e usa endereçamento para comunicação entre os dispositivos. Pode ter até 128 dispositivos em rede, permitindo mais de um dispositivo máster controlando o barramento. Tem controle de colisão e arbitragem para uso do barramento pelos dispositivos máster. Cada dispositivo tem seu endereço único. Originalmente usado só em equipamentos eletrônicos, as empresas fabricantes vem desenvolvendo circuitos integrados para adaptá-lo para outras situações, a exemplo dos extensores que pode levar o barramento I²C a algumas centenas de metros.

Após levantamento dos diversos protocolos de comunicação serial usados em rede de computadores e automação em geral, escolhemos alguns para comparação, que por suas características poderiam ajudar a compor um cenário de computação ubíqua. Eles estão representados na tabela 2.

Características	RS232	RS422	RS485	X10	USB	SPI	I²C
Velocidades	(19,6 kbps)	(19,6 kbps)	125 kHz	-	12MHz 480MHz	110kHz	100kHz 400kHz 3,4MHz
Custo	baixo	médio	alto	baixo	baixo	baixo	médio
Max. Dispositivos	1	10	32	256	127	32	128
Conexões	pp	pp	mp	pp	pp	mp	mp
Cabeamento	4 fios	4 fios	4 fios	-		2 + 1 fio	2 fios
Distâncias (metros)	15 m	1200 m	1200 m	**	5 m	-	*

pp = ponto-a-ponto mp = multi-ponto * limitada pela capacitância ** da rede

Tabela 2 - Comparativo entre os protocolos seriais

4.2 Tecnologias sem fio

Dos protocolos de tecnologias sem fio, que foram estudados e tiveram suas características comparadas, apresentamos aqui três que por seus requisitos para aplicação na automação residencial, foram consideradas eficientes. As duas primeiras foram escolhidas principalmente pelas suas características de cobertura. São as tecnologias Bluetooth [16] e Zigbee [17], esta última também conhecida como homeRF. É importante salientar que estas tecnologias não concorrem entre si e sim, se complementam num projeto de automação residencial.

A tecnologia Bluetooth permite a comunicação sem fio entre aparelhos eletrônicos que podem ser computadores, telefones celulares, PDA, equipamentos de escritório e dispositivos móveis. Um microchip muito pequeno, possuindo um transmissor de radio, é inserido em um dispositivo digital. A tecnologia Bluetooth realiza todas as conexões instantaneamente. Isso facilita uma rápida e segura transmissões de dados e voz, mesmo quando os dispositivos não estão em linha direta de visão. Esta comunicação realiza-se através de um dispositivo de enlace de rádio na freqüência de 2.4 GHz, que não necessita de licença e esta disponível em quase todo o mundo.

Bluetooth é mais apropriado para aplicações como:

- Sincronização de PC, telefone celular e PDA;
- Aplicações de áudio como fone sem fio;
- Transferência de arquivos entre PDA, PC e Impressoras.

Zigbee é um padrão para comunicações sem fio de baixa potência destinado a aplicações envolvendo vários dispositivos como, por exemplo, sensores. Bastante versátil, com Zigbee é possível formar redes em diversas topologias mantendo as características importantes do protocolo como baixo tempo de acesso à rede, baixo tempo de ativação dos dispositivos escravos e baixo consumo. Trabalha na freqüência de 2.4 GHz.

ZigBee tem melhor performance em aplicações:

- De Controle;
- Rede de Sensores;
- Redes com muitos dispositivos;
- Com pequenos pacotes de dados;
- Onde consumo de bateria é crítico.

A tabela 3 destaca as principais diferenças entre estes dois protocolos.

Outra diferença relevante entre estes dois padrões é o tipo de alimentação dos dispositivos. Em aplicações Bluetooth, geralmente, os dispositivos são recarregados periodicamente, como celulares e PDAS, enquanto no padrão Zigbee, estes podem ser alimentados com pilhas alcalinas comuns e a expectativa de duração das mesmas é superior a 2 anos.

Analisando estas diferenças pode-se dizer que ZigBee e Bluetooth são duas soluções diferentes para resolver problemas em aplicações diferentes. Mesmo que ocorram pequenas modificações nestes padrões as características principais serão mantidas. As diferenças entre estes padrões são provenientes da arquitetura na qual eles foram desenvolvidos [16].

Características	ZigBee	Bluetooth
Padrão (MAC + PHY)	IEEE 802.15.4	IEEE 802.15.1
Taxa de Transferência	250 kbps	750 kbps
Corrente na Transmissão	30mA	40mA
Corrente em repouso	3uA	200uA
Conexões	Ponto a ponto	Multi-ponto
Alcance (metros)	25 - 60 m	10 m
Tempo de acesso a rede	30ms	3s
Tempo de transição dos escravos (dormindo para ativo)	15ms	3s
Tempo de acesso ao canal	15ms	2ms

Tabela 3 - Diferenças entre ZigBee e Bluetooth

A terceira tecnologia sem fio é a chamada etiqueta RFID (IDentificador por Radio Freqüência) [19]. É um pequeno transmissor passivo, isto é, sem alimentação, que quando aproximado de uma leitora apropriada, consegue transmitir informações, com a energia que recebe da leitora. Tem sido usada no comércio, como proteção contra roubo de mercadorias. Começa a ser usado como controle de acesso, autenticando seus usuários, quando este apresenta o cartão a uma leitora, a uma distancia de dois a cinco centímetros. Usa a faixa de freqüência de 125 khz. Com uma pequena fonte de energia, as etiquetas passam a ser ativas e devido a esta alimentação própria, podem ser lidas a alguns metros de distancias.

Estes cartões têm a Motorola como seu principal fabricante e possuem vários protocolos, entre eles o wiegand, o mag-stripe, o racs, citando aqui só alguns. O protocolo wiegand possui vários formatos ou estrutura. O formato de 26 bits utiliza dois bits para controle, sendo o primeiro e o último. Os fabricantes utilizam 8 bits, chamados de código de facilidades, com informação de quem fabricou aquele cartão e 16 bits ficam livres para registrar informação do seu usuário.

5. ESCOLHA E IMPLEMENTAÇÃO DOS PROTOCOLOS PARA AUTOMAÇÃO RESIDENCIAL

Para controlar os dispositivos sensores e atuadores que vão existir em cada cômodo da residência, inicialmente foi elaborado um esboço de um circuito elétrico, chamado de ponto de controle, onde um microcontrolador de maior porte faz o papel de órgão central. Estamos trabalhando com o

microcontrolador PIC16F88 da Microchip [20]. Este microcontrolador possui barramento I2C e RS232. Tem 16 portas que podem ser programadas como entradas ou saídas. Para os cômodos, onde teremos os pontos de automação, será colocado um circuito integrado GPIO (General Purpose In Out) da Philips, de código PCA9501 [21], que é um CI de propósito geral que trabalha com entradas e saídas, fazendo as conexões entre os sensores e atuadores. Com o uso do GPIO poderemos usar sensores e atuadores analógicos, que são mais baratos.

Como os circuitos integrados serão da Philips ou de suas associadas, não teremos dificuldades em criar um barramento para usar o protocolo I2C, onde todos os módulos poderão trocar informações, através do endereço único de cada componente desta rede. Para poder cobrir distancia maiores, utilizaremos os extensores, também da Philips, que trabalham com dois barramentos. Estes extensores, como o de código P82B96, trabalham com I2C à dois fios num barramento e com quatro fios, dois para transmissão e mais dois para recepção, no outro barramento. São estes pares trançados de transmissão e recepção que fazem com que o barramento possa se estender a alguma centenas de metros. Outros circuitos integrados foram colocados no barramento I2C para complementar as funcionalidades do dispositivo. A figura 1 abaixo mostra um esboço das idéias iniciais do projeto.

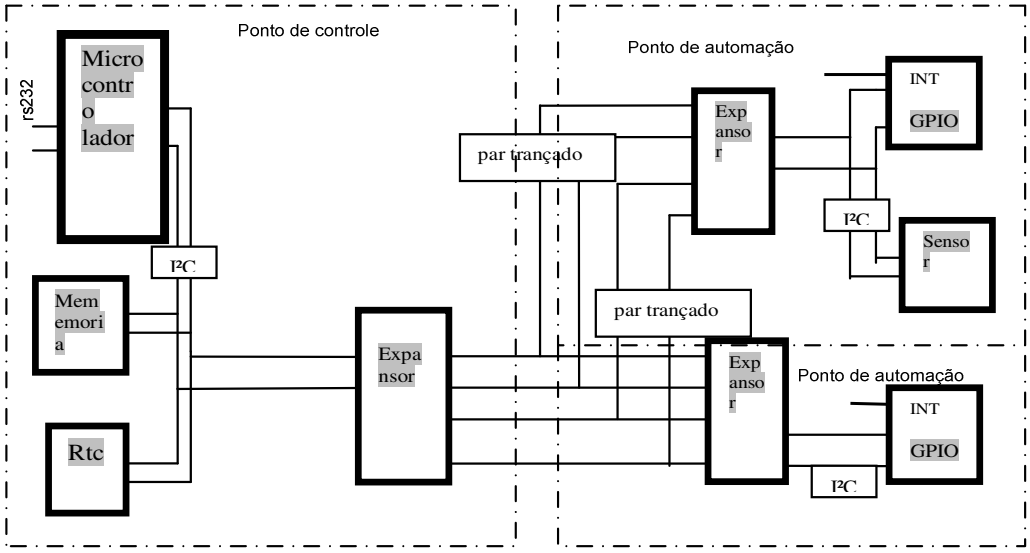


Figura 1 - Esboço inicial do projeto

Uma memória EEPROM, memória não volátil, no circuito integrado 24FC512 da Microchip [20], terá a função de armazenar, por exemplo, os cadastros com as autorizações de acesso solicitadas por um usuário. Se o computador que estiver fazendo a função de supervisório e servidor de autenticação estiver fora de funcionamento, a autenticação, que é a conferencia de informações para a autorização de acesso, poderá ser dada pelo microcontrolador, consultando a memória localizada no ponto de controle. Também, neste caso de falha do servidor de autenticação, é nesta memória que serão armazenadas as informações dos acessos permitidos ou negados. Quando o servidor voltar a funcionar, ele será atualizado com as informações mantidas na memória.

Um relógio calendário RTC (Real Timer Clock), de código PCF8583P da Philips [22], fornecerá a marcação do tempo para indexar os registros dos eventos que serão armazenados na memória. O armazenamento dos eventos com data e hora é importante para uso posterior em relatórios de auditorias, por exemplo.

No tocante a alimentação do circuito, utilizaremos tensão de 5 volts de corrente contínua para os circuitos integrados (microcontrolador, memória e extensor) e 12 volts de corrente contínua para as linhas de transmissão e recepção. Porém, o relógio será alimentado por uma bateria de 3 volts. Se houver problema com falta de energia, não precisaremos acertar o relógio para continuação do funcionamento do equipamento, na volta da energia.

Nos pontos de automação, além de contarmos com o GPIO fazendo a conexão com sensores e atuadores, podemos também adicionar circuitos integrados no barramento I2C local. Um exemplo seria colocar no barramento um sensor que trabalhasse diretamente com I2C.

Como o projeto de estudos dos protocolos precisa andar junto aos projetos da empresa pré-incubada, os engenheiros responsáveis pela QualiHouse, atendendo a encomenda de serviços de clientes, fizeram pequenas mudanças no andamento do projeto. Foi necessário desenvolver, implementar e testar um módulo de ponto de controle, para controle de acesso. Na figura 2 abaixo foi colocado um esboço do módulo que está sendo montado e os protocolos de comunicação empregados.

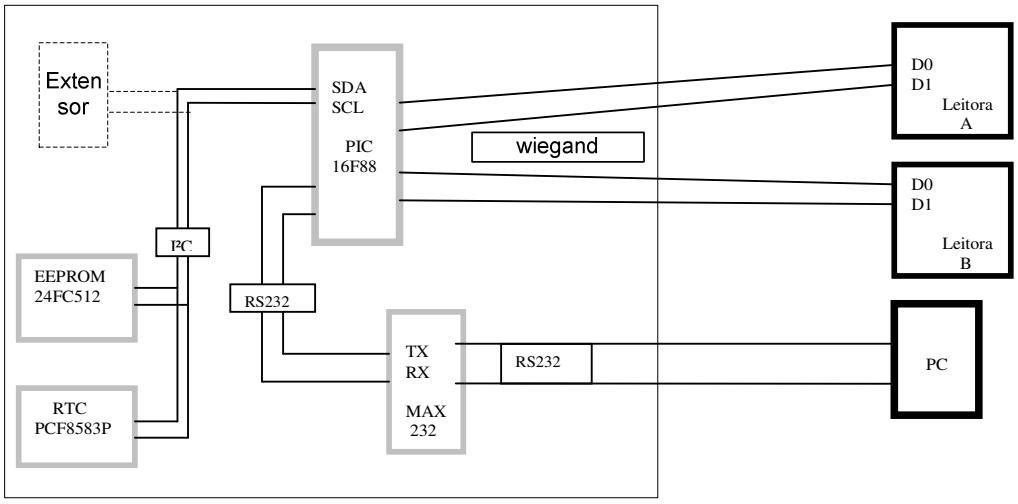


Figura 2 - Esboço atual do Projeto

Este módulo conta com duas leitoras de cartão RFID, uma para registrar acesso de entrada e outra para saída. Estas leitoras trabalham com o protocolo wiegand, no formato de 26 bits. Quando o usuário apresenta o cartão a leitora, esta faz a leitura, através de radio freqüência, das informações contidas no cartão e remete através de dois fios (D0 e D1) ao microcontrolador. Os bits são enviados assincronamente. Cada bit é enviado a um pulso de 50 micros segundos por 1 milisegundo de pausa. O microcontrolador, um PIC 16F88, trabalhando com clock interno de 8 MHz, recebe os dados da leitora e:

1º caso: envia ao computador através de sua interface RS232, usando o protocolo serial de mesmo nome, para fazer a autenticação do usuário do cartão. Se houver em seu banco de dados autorização para o acesso, o microcontrolador enviará comandos para que a porta seja aberta e sinalizando, através do led verde na leitora, do acesso permitido. Se não, acenderá o led vermelho sinalizando acesso negado.

2º caso: busca na memória EEPROM, através de sua interface I2C, caso o computador esteja por algum motivo fora de serviço, a autorização de acesso. Na memória EEPROM conterá dados suficientes dos usuários que possui permissão para acessar aquela porta controlada. As informações dos acessos permitidos e negados também deverão ser armazenadas na memória.

Os testes deste módulo de controle, com a montagem feita em um protoboard, foram executados e o resultado foi satisfatório. O pessoal da QualiHouse já mandou confeccionar a placa de circuito impresso, onde serão montados os componentes definitivos. Após montagem do circuito elétrico em placa definitiva, será feito a programação definitiva do software que será carregada no microcontrolador e os testes finais.

6. CONCLUSÃO

Um ambiente ubíquo deve possuir sistemas adaptativos, que acompanhem seus usuários, independente da sua localização, interagindo no novo contexto. Estes sistemas precisam armazenar o histórico dos hábitos de seus usuários para poder atendê-los. Os protocolos de comunicações para a automação destes ambientes devem ser escolhidos de acordo com os equipamentos existentes envolvidos ou os que ainda vão ser adquiridos para residência. Custo versus benefício deve ser perseguido, pois normalmente o usuário não deverá estar disposto a pagar pelo que não vai consumir, ou melhor, desfrutar.

Podemos considerar que os objetivos propostos para este projeto foram alcançados: aquisição de novos conhecimentos na área da computação ubíqua, suas necessidades e características; conhecimentos da automação residencial, dispositivos e fabricantes envolvidos na área e conhecimentos de protocolos diversos e suas aplicabilidades.

Devido a mudanças nos protótipos desenvolvidos pela Qualihouse, na segunda versão do projeto, como mostrado na figura 2, não foi ainda possível os testes com o relógio de tempo real. A QualiHouse ainda não adquiriu o componente, porém a placa de circuito impresso será confeccionada, considerando a sua existência futura.

Outro componente, o circuito integrado P82B96, extensor da rede I2C para distancias maiores, não será implantado nesta etapa. Um dos problemas é que o integrado que a QualiHouse recebeu de cortesia da Philips usa a tecnologia SMD (dispositivo de montagem de superfície). Para usar este tipo de circuito integrado precisamos criar uma placa suporte para recebê-lo, sem uso de solda, e conectar este suporte no protoboard. Estudos para esta finalidade estão em andamento. Já está pronto desenho do circuito da placa suporte.

Na seqüência destes estudos deveremos, após resolver o problema do extensor do barramento I2C, e testes com o relógio de tempo real, implementar as tecnologias sem fio. De início o bluetooth será empregado como controlador a distancia para acionar os dispositivos de abertura de portas. A tecnologia Zigbee deverá ser usada na implementação de um ponto de automação que poderá não ser alcançado pela tecnologia cabeada, como por exemplo, um sensor de presença numa área do lado de fora da residência.

7. REFERÊNCIAS

[1] **Computação Ubíqua.** Grupo de teleinformática e automação. UFRJ. Disponível em: http://recreio.gta.ufrj.br/grad/02_2/leandro/ Acessado em 03 de julho de 2006

[2] **Explorando Comportamento Sensível ao Contexto em Ambientes da Pervasive Computing**. Disponível: http://www.inf.ufrgs.br/~isam/IsamAdapt/index.html#Problema Acessado em 09 de setembro de 2005.

[3] Lara, S. M. A. **Computação Ubíqua**. Thaís Miranda Cia – Publicado em 13/06/2003. Disponível: http://coweb.icmc.usp.br/coweb/mostra.php?ident=95.5.3 Acessado em 01 de setembro de 2005.

[4] **Infra-estrutura de Suporte às Aplicações Móveis Distribuídas.** Projeto ISAM. Disponível: http://www.inf.ufrgs.br/~isam/index.html - Acessado em 31 de agosto de 2005.

[5] **Protocolos e Grupos de Trabalho.** Site AURESIDE – Associação Brasileira de Automação Residencial. Disponível: http://www.aureside.org.br/temastec/default.asp?file=protocolos03.asp Acessado 01 de setembro de 2005.

[6] Bettoni, R. L. **Automação residencial: um sonho que começa a se materializar.** Site Portal da Automação. Disponível em: http://www.portaldaautomacao.com.br/artigo_012.asp Acessado em: 03 de julho de 2006.

[7] **Cabeamento Estruturado**. Site AURESIDE - Associação Brasileira de Automação Residencial. Disponível: http://www.aureside.org.br/temastec/default.asp?file=infraestrutura.asp Acessado em 26 de agosto de 2005.

[8] **Peculiaridades dos Sistemas de Automação Residencial**. Site AURESIDE - Associação Brasileira de Automação Residencial. Disponível: http://www.aureside.org.br/temastec/default.asp?file=concbasicos02.asp – Acessado 01 de setembro de 2005.

[9] Strangio, C. E. **The RS232 Standard.** Disponível: http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html Acessado em: 12 de setembro de 2005.

[10] Smith, R. **QUICK REFERENCE FOR RS485, RS422, RS232 AND RS423** Disponível: http://www.rs485.com/rs485spec.html Acessado em: 12 de setembro de 2005.

[11] **Tecnologia de Portadora por Linha de Energia X10.** Disponível: http://www.aureside.org.br/temastec/x10.doc - Acessado em 26 de agosto de 2005.

[12] **Protocolos.** Disponível: http://69.56.194.34/~aterroso/html/protocolos.html - Acessado em 30 de setembro de 2005.

[13] **USB and IEEE** Disponível: http://www.semiconductors.philips.com/acrobat_download/literature/9397/75009799. pdf Acessado em 30 de setembro de 2005.

[14] **I2C Bus Technical Overview** Disponível: http://www.mcc-us.com/I2CBusTechnicalOverview.pdf - Acessado em 06 de setembro de 2005.

[15] **THE I2C - BUS SPECIFICATION** Disponível: http://www.semiconductors.philips.com/acrobat_download/literature/9398/39340011. pdf Acessado em 14 de outubro de 2005.

[16] **BlueTooth.** Soluções sem fio - Disponível: http://recreio.gta.ufrj.br/grad/00_2/bluetooth/INDEX.htm Acessado em 24 de outubro de 2005.

[17] **Artigos sobre ZigBee.** Disponível: http://www.eletronica.org/modules.php?name=News&file=article&sid=226 Acessado em 24 de outubro de 2005.

[18] **Data sheet** Disponível: http://www.atmel.com/dyn/resources/prod_documents/doc4681.pdf Acessado em:

[19] **Leitos RFID**. Site Saber Eletrônico. Disponível: http://www.sabereletronica.com.br/edicoes/edicoes.asp?comando=403_58&dettaglio=403 Acessado em: 3 de junho de 2006.

[20] **Product Microchip.** Disponível: http://www.microchip.com/stellent/idcplgidcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010243 Acessado em: 30 de setembro de 2005.

[21] **Product description**. Disponível: http://www.semiconductors.philips.com/pip/PCA9501.html Acessado em 30 de setembro de 2005.

[22] **Product description.** Disponível: http://www.semiconductors.philips.com/pip/PCF8583_5.html Acessado em: 30 de setembro de 2005.

Requerimientos de Qos en la Asignación Óptima de Lightpath Un enfoque multiobjetivo basado en MOEAs.

Carlos Núñez, Norihiko Isawa
Investigación Informática, Universidad Nacional de Itapúa
Campus Universitario Encarnación, Paraguay
{cnunez, norihiko}@uni.edu.py

Abstract

The intense traffic originated by applications in high performance networks forces the adoption of optical fibers transmission systems. Communications in optical networks along with wavelength-division multiplexing (WDM) technology, has contributed to increase available bandwidth. Suitable selections of lightpaths are expected with the utilization of Algorithms designed specifically for Routing and Wavelength Assignment (RWA). From the perspective of Quality of Service (QoS) in an all-optical network, the need for survivability and protection under eventually failure of a link could be expressed as a restoration path and a dedicated or shared path, according to the QoS of the request. This paper presents a version of Evolutionary Algorithms based on the Strength Pareto Evolutionary Algorithm – SPEA, and the Elitist Non-dominate Sorting Genetic Algorithm II – NSGA-II, for the optimal assignation of lightpath in survivable WDM networks. These two algorithms simultaneously minimized the wavelength conversion and the hop count for a set of given demand, considering QoS and wavelength constraint. Experiments carried out indicate the performance level of the proposed algorithms under different environmental situations, from low resources load to very high resources utilization.

Keywords: Computer networks, Optical Networks, Routing and Wavelength Assignment, Quality of Service, MultiObjective Optimization.

Resumen

El intenso tráfico generado por aplicaciones en redes de alto desempeño, hace que la utilización de fibras ópticas (FO) sea un imperativo. La transmisión en redes de FO con la mutiplexación por división de longitud de onda ha contribuido al aumento de la capacidad de ancho de banda utilizable. Se espera además un óptimo desempeño con la selección de *lightpath* mediante algoritmos especializados en las tareas de enrutamiento y Asignación de longitudes de onda (*Routing and Wavelength Assignment* – RWA). Desde el punto de vista de la Calidad de Servicio (QoS) en redes ópticas se introducen algunas formas de recuperación y/o protección ante caídas eventuales en los enlaces, por medio de asignación de *lightpaths* alternativos a los ya existentes, los cuales pueden ser compartidos o dedicados dependiendo del QoS. Se propone una solución basada en Algoritmos Evolutivos MultiObjetivos (*MultiObjective Evolutionary Algorihtms* – MOEAs), específicamente el SPEA y el NSGA-II para la asignación óptima de *lightpaths* con requerimientos de QoS en redes WDM con capacidad de recuperación, y minimizar el número de conversiones de longitudes de onda (*wavelength conversion*) y la cantidad de saltos en forma simultánea (*hop count*) con requerimientos de QoS. Resultados experimentales muestran el desempeño de los algoritmos en situaciones con diversos grados de complejidad, desde poca carga hasta el límite de utilización extrema de recursos.

Palabras Claves: Redes de computadoras, Redes Ópticas, Enrutamiento y Asignación de Longitud de Ondas, Calidad de Servicio, Algoritmos Evolutivos, Optimización Multiobjetiva.

1 INTRODUCCIÓN

El tráfico intenso y creciente en las redes de comunicaciones actuales, como Internet 2, exige más recursos para satisfacer las necesidades de transmisión de información. Este objetivo puede alcanzarse con la utilización de WDM (*Wavelength Division Multiplexing*) en redes de fibras ópticas, dividiendo el ancho de banda en porciones más pequeñas superando el problema de cuello de botella electrónico (*electronic bottleneck*) [1]. De esta forma, múltiples flujos de datos pueden ser transmitidos en forma simultánea sobre una misma fibra óptica. El enrutamiento en canales ópticos y el soporte de conmutación proveen una amplia gama de señales con las que se puede transmitir a velocidades del orden de los 40 Gbps [2]. Redes puramente ópticas que utilizan WDM y *Wavelength Routing* son soluciones totalmente viables para distintos tipos de redes como las de áreas extensas y de áreas metropolitanas. Una conexión de un nodo a otro en cualquiera de estas redes a través de una longitud de onda particular, es denominada *lightpath*, y se cuenta con algoritmos apropiados de enrutamiento y asignación de longitudes de ondas (*Routing and Wavelength Assignment* – RWA) [2]. En ausencia de convertidores de longitudes de onda, la misma debe ser mantenida a lo largo de todo el *lightpath*, regla conocida como restricción de continuidad [3]. La probabilidad de no satisfacer una demanda, en estas circunstancias, resulta más crítica que en las redes que utilizan convertidores de longitudes de onda.

En problemas de RWA, se presentan generalmente dos escenarios típicos. *1) Asignación estática:* cuando todos los *lightpaths* son definidos por adelantado[3]; *2) Asignación dinámica:* cuando los *lightpaths* son creados bajo demanda, y al finalizar se liberan todos los recursos (canales WDM), quedando éstos disponibles para futuras conexiones. A efectos de proveer supervivencia a las conexiones (*survivability*), se han propuesto recientemente varios paradigmas ampliamente estudiados y bien definidos, tanto para asignación dinámica como estática, como ser: *1)* esquemas *basados en protección* [5, 6 y 7] donde, dos caminos (o *lightpath*) se crean para el mismo par origen-destino, siendo estos disjuntos y uno de ellos es utilizado como backup, y *2)* esquemas *basados en restauración* [8 y 9] donde, un nuevo camino se calcula en tiempo real cuando se detecta una falla en el lightpath actual (primario). En el esquema basado en protección se pueden distinguir también dos enfoques: *a)* configuraciones dedicadas (*dedicated path),* donde el camino alternativo no puede ser utilizado por otra demanda y *b)* configuración compartida (*shared path*), donde los recursos reservados en caso de fallas pueden compartirse utilizando algún esquema de prioridades. Zhong y Jaeke propusieron aplicar el concepto de esquemas de protección para lograr múltiples niveles de Calidad de Servicio (QoS), donde un lightpath alternativo es asignado como backup dedicado o compartido según los requerimientos[4]. Dicho trabajo realiza una optimización mono-objetiva del número de saltos con niveles de QoS. En contrapartida, este trabajo propone la utilización de Algoritmos Evolutivos MultiObjetivos (*MultiObjective Evolutionary Algorithms* – MOEAs), basadas en población de soluciones, exitosamente utilizadas en varios problemas de la vida real, como el enrutamiento tradicional [10] entre otros. Consecuentemente, este trabajo implementa dos reconocidos algoritmos evolutivos multiobjetivo de segunda generación, el *Strength Pareto Evolutionary Algorithm* (SPEA) y el *Elitist Non-dominated Sorting Genetic Algorithm* (NSGA-II) para el problema de asignación de *lightpath* bajo requerimientos de QoS en redes WDM. Esta propuesta minimiza en forma simultánea, el número de saltos y el número de conversiones de longitudes de onda de un conjunto conocido de solicitudes unicast, bajo restricciones impuestas por los múltiples niveles de QoS.

El resto del trabajo se organiza de la siguiente forma: en la Sección 2 se delinean los conceptos básicos de calidad de servicio o QoS, mientras que la definición formal del problema de optimización multiobjetivo se presenta en la Sección 3. En la Sección 4 se da la formulación del

problema. En la Sección 5 se expone el enfoque de resolución basado en MOEAs, mientras que en la Sección 6 se sintetiza al ambiente experimental y los resultados. Finalmente en la Sección 7 se presentan las conclusiones y trabajos futuros.

2 CALIDAD DE SERVICIO EN REDES ÓPTICAS

QoS, en términos de Ingeniería de Tráfico, se refiere a la capacidad de proveer diferentes prioridades a diferentes tipos de usuarios o flujos de datos, o garantizar un cierto nivel de rendimiento a un flujo, en concordancia con los requerimientos de cada aplicación. En una red de computadoras, QoS puede representar mecanismos avanzados para influir en la probabilidad de proveer el nivel solicitado de rendimiento. Según Tanenbaum [11], los parámetros principales determinan la QoS de cada flujo son : confiabilidad, retardo, fluctuación y ancho de banda.
En redes ópticas WDM, la asignación de un *lightpath* bajo las restricciones de utilización de longitudes de onda (*wavelength constraint*) minimiza la problemática de retardo y la fluctuación originada por congestionamiento. Normalmente, las redes ópticas WDM son propensas a catastróficas caídas de enlaces, debido a altos volúmenes de tráfico, por razones que incluyen las fallas de fibras ópticas (que comúnmente ocurren cada 4 días en redes de gran porte [2]), transmisores, receptores, conversores entre otros. Una falla de enlace es descubierta fácilmente cuando los nodos detectan la ausencia de algún haz de luz sobre el enlace. Un algoritmo de administración de redes es invocado para notificar del evento y entonces se llevan a cabo los procedimientos necesarios para la recuperación de la comunicación.
Así, la habilidad de reconfigurar y re-establecer la comunicación al momento de la falla debe ser aprovisionada al momento de establecer una conexión [2]. Teniendo en cuenta que la caída de alguna conexión puede deberse a varias causas, los tipos de fallas se clasifican en: *1)* fallas de un solo enlace (*single-link failure*) o *2)* fallas de múltiples enlaces (*multi-link failure*). En este trabajo, para el cálculo del camino alternativo (o de backup) se considera que la probabilidad de caída de múltiples enlaces es despreciable.
Por otra parte, la provisión de un *lightpath* alternativo dedicado obliga a tener recursos extras asignados a una demanda de alta prioridad. Estos recursos son costosos en extremo por lo que es indispensable también considerar esquemas compartidos para aplicaciones menos críticas. En ese sentido, Zhong y Jaeke [4] proponen un esquema híbrido en el que se ofrecen protección dedicada, compartida y una modalidad sin protección según la calidad de servicio QoS, requerida por la aplicación en cuestión. Un esquema de protección para transmisiones unicast, semejante al propuesto por Zhong y Jaeke [4], es utilizado en este trabajo. Básicamente se define tres tipos de QoS:

- *ORO*: La calidad más alta representada en la jerarquía. Una solicitud al ser procesada, obtiene un camino principal y otro alternativo, cuyos componentes son disjuntos. Ambos *lightpaths* son exclusivos y corresponden al esquema de protección dedicada.
- *PLATA*: Similar a la categoría ORO. Durante el proceso de atención a la solicitud se asignan dos caminos disjuntos entre ellos, uno como primario y el otro como camino alternativo. A diferencia de la clase ORO, su camino alternativo puede ser utilizado para crear nuevas conexiones primarias para otras solicitudes de menor calidad. Esto pone en práctica al esquema de protección compartida.
- *BRONCE*: La de menor calidad dentro de la jerarquía. Las solicitudes con este tipo serán atendidas solamente con un camino principal (sin camino alternativo o de backup, *best effort*).

Es importante destacar, que si una respuesta de calidad Bronce utiliza algún componente del lightpath secundario de otra solicitud de calidad Plata, la caída del primario de ésta última solicitud fuerza a interrumpir la transmisión de la solicitud Bronce, por ser esta última de menor prioridad.

3 OPTIMIZACIÓN MULTIOBJETIVO

La optimización multiobjetivo [15] consiste en hallar soluciones que optimicen *k* objetivos posiblemente conflictivos entre sí y puede ser formulado como el problema de encontrar un vector de decisión $\boldsymbol{x}$ en el espacio de decisión $\boldsymbol{X}$ que satisfaga *m* restricciones y optimice un vector de objetivos *y* en el espacio objetivo $\boldsymbol{Y}$, conforme se expresa a continuación:

$$\begin{aligned} &\text{Optimizar } \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}),\ldots, f_k(\boldsymbol{x})]^T \in \boldsymbol{Y} \\ &\text{Sujeto a } \quad \boldsymbol{g}_i(\boldsymbol{x}) \geq 0, \quad i = 1,\ldots, m. \\ &\text{donde } \quad \boldsymbol{x} = [x_1, x_2,\ldots,x_n]^T \in \boldsymbol{X}. \end{aligned} \tag{1}$$

Considerando varios objetivos a optimizar, no siempre es posible una tradicional comparación unidimensional que indique si una solución es mejor que otra. En consecuencia se utiliza el conocido concepto de dominancia Pareto [15], en el que un vector objetivo $\boldsymbol{y}_1$ domina a otro $\boldsymbol{y}_2$ ($\boldsymbol{y}_1 \succ \boldsymbol{y}_2$) si y solo si $\boldsymbol{y}_1$ no es peor que $\boldsymbol{y}_2$ en ningún objetivo y es estrictamente mejor que $\boldsymbol{y}_2$ en al menos un objetivo. Del mismo modo, podemos decir que una solución $\boldsymbol{x}_1$ es mejor que otra $\boldsymbol{x}_2$ si $\boldsymbol{f}(\boldsymbol{x}_1) \succ \boldsymbol{f}(\boldsymbol{x}_2)$, se dice entonces que $\boldsymbol{x}_1$ es mejor que $\boldsymbol{x}_2$ o que $\boldsymbol{x}_1$ domina a $\boldsymbol{x}_2$. En el caso que ni $\boldsymbol{x}_1$ domine a $\boldsymbol{x}_2$ ($\boldsymbol{x}_1 \not\succ \boldsymbol{x}_2$) ni $\boldsymbol{x}_2$ domine a $\boldsymbol{x}_1$ ($\boldsymbol{x}_2 \not\succ \boldsymbol{x}_1$) se dice que las soluciones no son comparables, lo que se denota como ($\boldsymbol{x}_1 \sim \boldsymbol{x}_2$).
Las características hasta ahora expuestas hacen que sea posible tener un conjunto de soluciones de compromiso, compuesto por todas las soluciones no dominadas del dominio. A este conjunto de soluciones óptimas se denomina conjunto Pareto (P_{true}) y su respectiva imagen en el espacio objetivo se conoce como frente Pareto (Y_{true}).
En el contexto de este trabajo se consideran solo problemas de minimización, sin que esto implique una pérdida de generalidad en la formulación matemática.

4 FORMULACIÓN MATEMÁTICA

Para este trabajo, una red de comunicaciones ópticas es modelada como un grafo $G = (V, E, C)$, donde V es el conjunto de nodos, E es el conjunto de enlaces y C es el conjunto de longitudes de onda disponibles en cada enlace óptico perteneciente a E. Sea:

$(i, j) \in E$	enlace óptico desde el nodo i al nodo j; donde $i, j \in V$;
(i, j, λ)	canal óptico desde el nodo i al nodo j, con longitud de onda λ, donde $i, j \in V$ y $\lambda \in C$;
c_{ij}	número máximo de longitudes de onda disponibles en el enlace (i, j);
Q	conjunto de calidades de servicio ofrecidos por la red, en este trabajo Q = {Oro, Plata, Bronce};
$u= (f, d, q)$	solicitud unicast u con nodo fuente f y nodo destino d, donde $f, d \in V$ y $q \in Q$;
p_u	*lightpath* primario correspondiente a la solicitud unicast u;
b_u	*lightpath* alternativo o *backup* correspondiente a la solicitud unicast u;
$s_u=(p_u, b_u)$	respuesta a solicitud u, con *lightpath* primario y alternativo atendiendo al requerimiento $q_u \in$ Q;

U	conjunto de solicitudes unicast, con $U = \{u_1, u_2, \ldots, u_{\|U\|}\}$, donde $\|.\|$ indica cardinalidad de un conjunto;
u_{ij}^{λ}	longitud de onda (λ) asignada a la solicitud unicast u en el enlace (i, j);
S_U	solución atendiendo al conjunto de solicitudes unicast U, i.e. $S_U = \{s_{u1}, s_{u2}, s_{u3}, \ldots, s_{u\|U\|}\}$.

Utilizando las definiciones anteriores, el RWA bajo requerimientos de QoS puede ser tratado como un problema multi-objetivo (*Multi-Objective Problem* - MOP) que busca la mejor solución S_U que simultáneamente minimice las siguientes funciones objetivos:

1- *Número total de saltos:*

$$y_1 = \sum_{u \in U} \left(\sum_{(i,j) \in s_u} \phi_{ij} \right) \qquad \text{donde:} \quad \phi_{ij} = \begin{cases} 1 & \text{si } (i,j) \in s_u \\ 0 & \text{en otro caso} \end{cases} \tag{2}$$

2- *Número de conversiones de longitud de onda:*

$$y_2 = \sum_{u \in U} \left(\sum_{i \in V} \varphi_{ij} \right) \qquad \text{donde:} \quad \varphi_{ij} = \begin{cases} 1 & \text{si en el nodo } i \in V \text{ se conmuta } \lambda \text{ para la solicitud } u \\ 0 & \text{en otro caso} \end{cases} \tag{3}$$

Bajo las siguientes restricciones:

- Dada la solicitud $u = (f, d, q)$ tal que $q \in$ {Oro, Plata}, se tiene que: *Si un enlace óptico* $(i, j) \in p_u \rightarrow (i, j) \notin b_u$ y *Si un nodo* $i \in p_u$ con $i \neq f, d \rightarrow i \notin b_u$, i.e. se trata de un esquema de protección a nivel de path, por lo que un enlace (i, j) del primario ya no puede ser usado para el backup.
- Dada la solicitud $u = (f, d, Oro)$, se tiene que: *Si un canal óptico* $(i, j, \lambda) \in \{p_u, b_u\} \rightarrow (i, j, \lambda) \notin S_U - \{p_u, b_u\}$. i.e. se trata de un esquema dedicado de protección donde el uso del canal óptico (i, j, λ) inhabilita su uso posterior.
- Dada la solicitud $u = (f, d, q)$ tal que $q \in$ {Oro, Plata, Bronce}, se tiene que: *Si un canal óptico* $(i, j, \lambda) \in p_u \rightarrow (i, j, \lambda) \notin S_U - \{p_u\}$, i.e. se trata de una restricción en la utilización de un canal óptico (i, j, λ) que se utiliza en un *lightpath* primario. Se lo puede interpretar como una restricción en la re-utilización de una longitud de onda (*wavelength constraint*) ya asignada.

Para poder aclarar los conceptos presentados y denotar las características de la calidad de servicio en redes ópticas, el siguiente ejemplo expone un problema sencillo y una posible solución.

Ejemplo 1. Dada la red con la topología NSFnet [13] en la Figura 1, con 14 nodos y 42 enlaces, se solicita acomodar el siguiente conjunto de demandas unicast con sus requerimientos QoS respectivos $U = \{u_1 = (0, 13, Oro), u_2 = (4, 13, Plata), u_3 = (0, 8, Bronce)\}$. En la Figura 1.a puede visualizarse la asignación del *lightpath* primario p_{u1} y alternativo b_{u1} disjuntos para la solicitud u_1. Los lightpath primario p_{u2} y alternativo b_{u2} disjuntos para la segunda solicitud u_2 son presentadas en la Figura 1.b. La asignación para la solicitud u_3 es esquematizada en la Figura 1.c, aquí los requerimientos de QoS son mínimos por tanto "no" se asigna un *lightpath* alternativo.

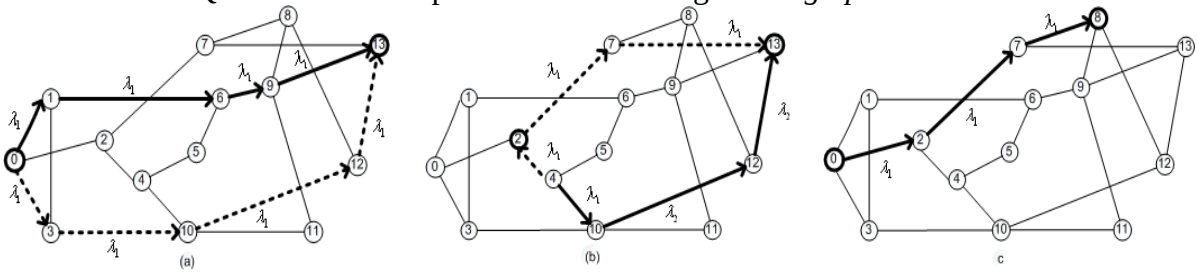


Figura 1. Asignación de *lightpath* para el Ejemplo 1. a) Path primario y secundario para la u_1 con QoS Oro. b) Solicitud u_2 con QoS Plata. c) Asignación u_3 con requerimientos de baja QOS "no" se asignan *lightpath* alternativos o de backup.

Note que el *lightpath* alternativo b_{u2} comparte el mismo canal óptico (2, 7, λ_1) con el *lightpath* p_{u3}, dado que el requerimiento de calidad de servicio de u_2 asume que su camino alternativo puede ser utilizado por otras demandas, mientras no exista alguna falla p_{u2}. Esto implica que si el *lightpath* b_{u2} es requerido por la solicitud u_2, la transmisión de datos de u_3 se vera interrumpida. Para esta posible solución S_U, el correspondiente vector objetivo es $\boldsymbol{y} = (y_1, y_2) = (17, 1)$, i.e. son necesarios 17 enlaces ópticos y solo una conversión de longitud de onda en el nodo 10 para el *lightpath* b_{u2}.

5 ALGORITMOS EVOLUTIVOS MULTIOBJETIVOS

Investigaciones realizadas sobre el principio de *survival of the fittest* (ley de supervivencia de los más aptos) observado en la naturaleza, dieron como resultado simulaciones computacionales que resultaron muy útiles para solucionar problemas complejos [14], dando origen a los Algoritmos Evolutivos (*Evolutionary Algorithms* – EAs). Los EAs han sido muy populares en tareas de búsqueda y de optimización en los últimos años, con un desarrollo constante de nuevos algoritmos. En particular, los Algoritmos Evolutivos MultiObjetivos (*MultiObjective Evolutionary Algorithms* – MOEAs) permiten resolver problemas de optimización multiobjetivos (MOP), encontrando un conjunto completo de soluciones *Pareto* en una sola ejecución [15], convirtiéndolos en un candidato natural para resolver el problema del enrutamiento óptico. En la literatura, se tienen trabajos que comparan el rendimiento de estos Algoritmos. En [16] se analiza el rendimiento de NSGA-II, SPEA-2 y CNSGA-II con elitismo controlado en diseño de sistemas de seguridad. En [22] el SPEA, SPEA-2, NSGA-II y CNSGA-II fueron comparados en el problema de enrutamiento multicast multiobjetivo, donde los resultados indican que no existe una clara superioridad de algún enfoque.
Es este trabajo fueron implementados el SPEA y el NSGA-II, enfoques evolutivos de significativa simplicidad pero de notable eficacia, como un primer acercamiento de los MOEAs al problema de asignación óptima de *lightpath* con características de QoS.

5.1 Strength Pareto Evolutionary Algorithm – SPEA

Este algoritmo fue introducido por el Zitzler y Thiele [18] en el año 1999. El SPEA utiliza un archivo que contiene las soluciones no dominadas encontradas (población externa de no dominados P_{nd}). En cada generación, se copian los individuos no dominados de P a P_{nd} y se borra de ésta última las soluciones dominadas. Para cada individuo en el sistema externo, se computa un valor de fuerza (*strength*) que es proporcional al número de soluciones a las cuales cada individuo domina. Por otra parte, el *fitness* de cada miembro de la población actual P se computa según las fuerzas de todas las soluciones no dominadas externas que la dominen [18].

5.2 Elitist Non-dominated Sorting Genetic Algorithm – NSGA-II

El algoritmo NSGA-II fue propuesto por Deb et al. [19]. Se basa en la clasificación eficiente de individuos en varias capas o frentes (F_i). La clasificación consiste en agrupar a todos los individuos no dominados en un frente, con un valor de *fitness* (o adaptabilidad) calculado a través de una distancia de *Crowding* [19]. Este valor es proporcional a la distancia de cada objetivo de los vecinos más próximos del mismo frente al que pertenece. Entonces el grupo de individuos clasificados es ignorado y otro frente de individuos no dominados es considerado. El proceso continúa hasta que se clasifican a todos los individuos en la población. En el proceso de selección de un individuo, se

considera un orden lexicográfico, donde se tiene primero en cuenta el índice del frente al cuál pertenece y luego la distancia de *Crowding*. Los Algoritmos 1 y 2 sintetizan el enfoque del SPEA y NSGA-II respectivamente

Algoritmo 1: SPEA

1: Inicializar población inicial P; t = 0
2: **mientras** criterio de parada no se cumpla **hacer**
3: Evaluar individuos de P_t
4: Copiar los individuos no dominados de P_t a P_{nd}
5: Eliminar las soluciones de P_{nd} que sean dominadas por otras soluciones de P_{nd}
6: Calcular el fitness de cada uno de los individuos en P_t así como en P_{nd}
7: Si el número de soluciones almacenadas en P_{nd} superan un máximo establecido N, reducir P_{nd} por medio del clustering
8: Seleccionar individuos del conjunto P_t y P_{nd} (*mating pool*)
9: Aplicar los operadores de cruzamiento y mutación al *mating pool.*
10: Copiar nuevas soluciones del *mating pool* en $P_{t+1;}$ t = t+1
11: **fin mientras; imprimir** P_{nd}

Algoritmo 2: NSGA-II

1: Inicializar una población inicial P
2: Evaluar individuos de P y ordenar considerando dominancia
3: Aplicar operadores genéticos a P para obtener población Q; t = 0
4: **hacer**
5: $R = P_t \cup Q_t$ /* P_t representa la población P en la generación t
6: Ordenar R, considerando dominancia y obtener frentes F_i
7: l = 1
8: **mientras** $|P_{t+1}| < N$ /* N es el tamaño máximo de la población P y Q
9: Calcular adaptabilidad de cada individuo en F_i
10: $P_{t+1} = P_{t+1} \cup F_i$
11: l = l +1
12: **fin mientras**
13: Ordenar P_{t+1} por dominancia
14: Elegir los primeros N elementos de P_{t+1}, para tener Q_{t+1}
15: t = t + 1
16: **mientras** criterio de parada no sea alcanzado
17: **imprimir** Q

5.3 Cromosoma y Operadores Evolutivos.

Los Algoritmos Evolutivos son enfoques de propósito general. Su funcionamiento y metodología de resolución son prácticamente independientes al problema de optimización a ser resuelto. Para poder aplicar un Algoritmo Evolutivo a un problema en particular es necesario definir *1)* la estructura del cromosoma, que básicamente representa una solución potencial al problema considerado y *2)* un operador de cruce acorde al cromosoma. En este trabajo fue utilizado como proceso de selección el torneo binario por ser simple pero de alta eficiencia [20]. No se utiliza un operador de mutación, debido a las características aleatorias del cruce implementado. Seguidamente, se muestra la representación del cromosoma como también la del operador de cruce para el problema en cuestión.

Cromosoma. El problema de asignación de *lightpath* con requerimientos de QoS es representado como un vector de *lightpaths* primarios y alternativos como puede notarse en la Figura 2.
El cromosoma representa básicamente, a una solución $S_U = \{s_{u1}, s_{u2},\ldots, s_{uk},\ldots, s_{u|U|}\}$, donde cada $s_{uk} = (p_{uk}, b_{uk})$. De la definición matemática se observa que p_{uk} y b_{uk} representan los *lightpaths* primario y alternativo respectivamente (Sección 4). En la Figura 2 es presentada también la estructura del p_{uk}, análoga a la estructura de b_{uk}.

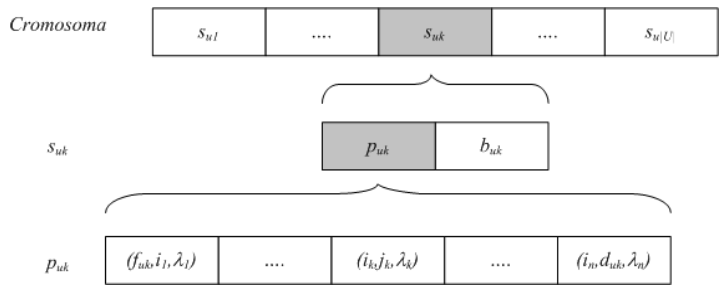


Figura 2. Relación de un cromosoma con los lightpath primarios y alternativos.

Cruce. En el cruce intervienen dos padres seleccionados a través del torneo binario, para obtener un solo individuo (hijo). El cruce es realizado a nivel de *lightpath*, solo en los caminos primarios para cada solicitud unicast del cromosoma. El cruce propuesto consiste en copiar todos los enlaces coincidentes directamente a la nueva solución, independientemente al tipo de longitud de onda. Finalmente, un algoritmo de enrutamiento estocástico basado en el "*Shortest Path Dijkstra* (SPD) [21]" asigna los restantes enlaces para completar el camino primario. Luego de obtener el camino primario, se procede a aplicar el SPD para obtener el camino alternativo. La asignación de longitudes de ondas es realizada mediante un algoritmo aleatorio que intenta evitar la conmutación de longitudes de ondas. En la Figura 3 es presentado un bosquejo del operador de cruce aplicado al k-esimo *lightpath* primario p_{uk} de dos cromosomas padres (A y B).
Se observa que los enlaces (a, b) y (c, d_{uk}) son utilizados por ambos padres, por tanto el cromosoma hijo en el k-esimo *lightpath* hereda dichos caminos.

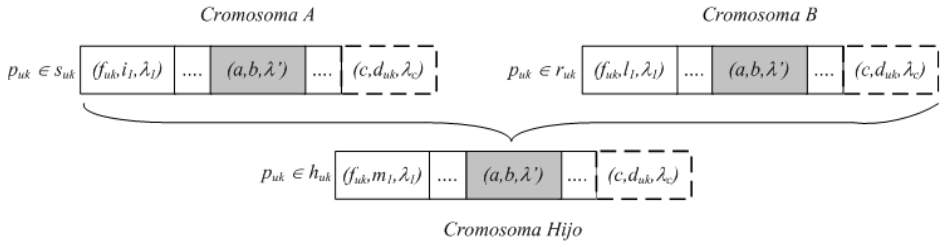


Figura 3. Operador de cruce aplicado a cada lightpath primario de los cromosomas padres.

6 AMBIENTE Y RESULTADOS EXPERIMENTALES

El SPEA y el NSGA-II fueron implementados en una PC Intel Core 2 Duo de 1,87 GHz con 1 GB de RAM, compilador javac 1.5.0_06 en un *Runtime Environment* versión 2.

Dos problemas de prueba fueron resueltos con el objeto de evaluar los métodos propuestos. Para cada uno fueron seleccionados varios conjuntos U de demandas unicast, con las mismas cantidades de longitudes de onda (λ). Los parámetros utilizados fueron: 50 cromosomas para la población evolutiva con 500 generaciones como criterio de parada para cada corrida. Un conjunto de soluciones aproximadas al Frente Pareto Real fue calculado para cada conjunto unicast U usando el siguiente procedimiento:

1. Cada algoritmo fue ejecutado 10 veces.
2. Un conjunto de soluciones formado por todas las soluciones de las 10 corridas fue obtenido.
3. Las soluciones dominadas fueron eliminadas, y un conjunto aproximado al Frente Pareto llamado "$\boldsymbol{Y}_{known}$" fue creado.
4. Luego, cada una de las 10 corridas fue comparada con $\boldsymbol{Y}_{known}$ para obtener promedios de soluciones aportadas.

6.1. Problema de Prueba 1

La topología NSFnet [13] con 14 nodos y 42 enlaces presentada en la Figura 1 fue utilizada en este primer problema. El conjunto de demandas unicast U utilizado es presentado en la Tabla 1. Note que la calidad de servicio (q) de cada solicitud es especificada con números, donde Oro ≡ 0, Plata ≡ 1, Bronce ≡ 2.

Tabla 1. Conjuntos de demandas unicast a ser acomodados en la NSFnet. La columna |*U*| indica la cantidad de solicitudes unicast para cada conjunto.

|*U*|	*U*
10	(0,13,2) (12,3,2) (2,11,2) (10,7,2) (6,12,2) (2,12,1) (10,6,1) (0,12,1) (1,12,0) (7,11,0)
20	(1,12,2) (6,13,2) (3,11,2) (0,8,2) (4,11,2) (3,13,2) (1,13,2) (2,11,2) (8,6,2) (9,0,2) (5,13,2) (12,5,2) (7,5,1) (3,12,1) (12,4,1) (1,13,1) (1,13,0) (2,11,0) (6,13,0) (9,0,0)
30	(1,12,2) (6,13,2) (3,11,2) (0,8,2) (4,11,2) (3,13,2) (1,13,2) (2,11,2) (8,6,2) (9,0,2) (5,0,2) (12,5,2) (10,7,2) (2,9,2) (4,9,2) (5,13,2) (8,3,2) (10,8,2) (1,13,1) (3,12,1) (7,5,1) (10,0,1) (12,4,1) (4,3,1) (5,7,1) (1,13,0) (2,11,0) (6,13,0) (9,0,0) (7,12,0)
40	(1,12,2) (6,13,2) (3,11,2) (0,8,2) (4,11,2) (3,13,2) (1,13,2) (2,11,2) (8,6,2) (9,0,2) (5,0,2) (12,5,2) (10,7,2) (2,9,2) (4,9,2) (5,13,2) (8,3,2) (10,8,2) (8,12,2) (6,7,2) (0,4,2) (2,1,2) (5,8,2) (4,7,2) (1,13,1) (3,12,1) (7,5,1) (10,0,1) (12,4,1) (4,3,1) (5,7,1) (6,10,1) (2,9,1) (10,3,1) (1,13,0) (2,11,0) (6,13,0) (9,0,0) (7,12,0) (11,7,0)

En la Figura 4 se muestran los resultados de las comparaciones del promedio en 10 corridas con $\boldsymbol{Y}_{known}$, con 500 generaciones divididas de a 50 para observar el desempeño a lo largo de las corridas. El SPEA tuvo mejor desempeño que el NSGA-II en todos los conjuntos de solicitudes donde la carga era baja (U={10,20}) a excepción de las corridas con cantidad de generaciones bajas (50, 100 y en parte de la generación 200, figuras 4.a, 4.b y 4.c), a partir de (U={20}) se observa un aumento en el desempeño del NSGA-II, al incrementar la cantidad de generaciones, como se ve en las figuras 4.d al 4.f.

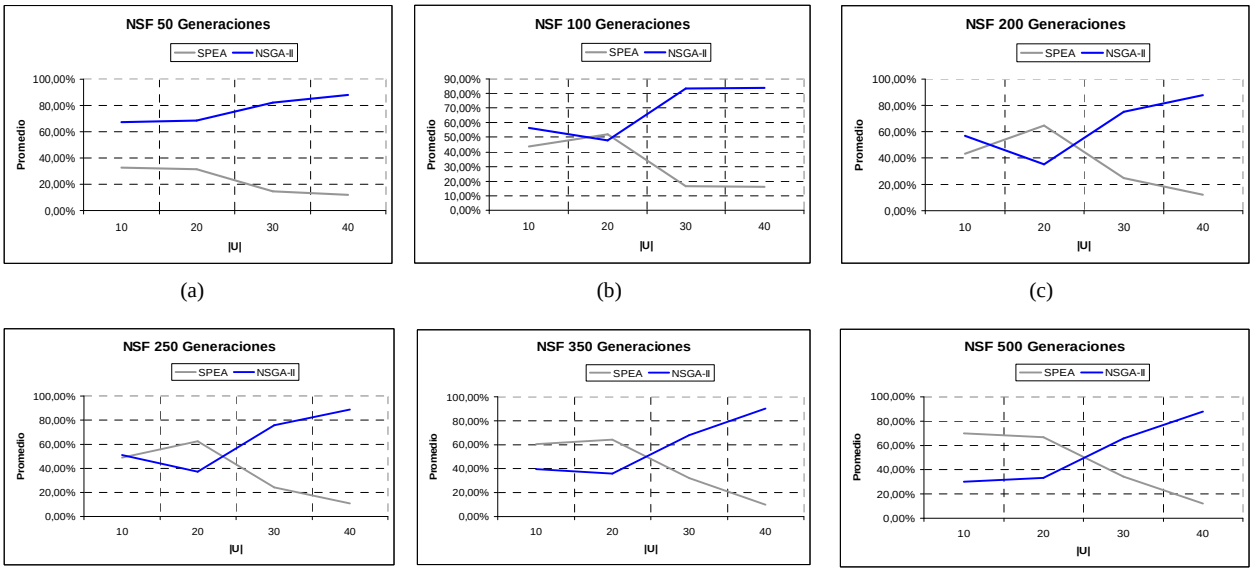


Figura 4. Resultados para el problema de prueba 1 sobre la topología de red NSFnet.

Los resultados arrojados por las pruebas del problema 1, indican que el SPEA es consistentemente mejor que el NSGA-II para todas las corridas llevadas a cabo donde la carga era baja, y las generaciones no superaban las 200. Sin embargo, teniendo en cuenta los resultados obtenidos en todas las generaciones, se verifica que aumentando la complejidad del problema la situación de dominancia se revierte, para la red NSF. Este resultado podría ser aún más contundente llevando cabo pruebas más complejas, como ser la utilización de una red cuya topología sea más densa. Teniendo en cuenta estos objetivos, se propone la utilización de la red NTTnet , como se presenta en la siguiente prueba.

Problema de Prueba 2

En el segundo problema fue utilizada la topología de prueba de la NTTnet [22]. NTTnet ya es de un tamaño considerable, con 55 nodos y 144 enlaces, conforme muestra la Figura 5. La Tabla 3 presenta los cuatro conjuntos de demandas unicast *U* con sus QoS.

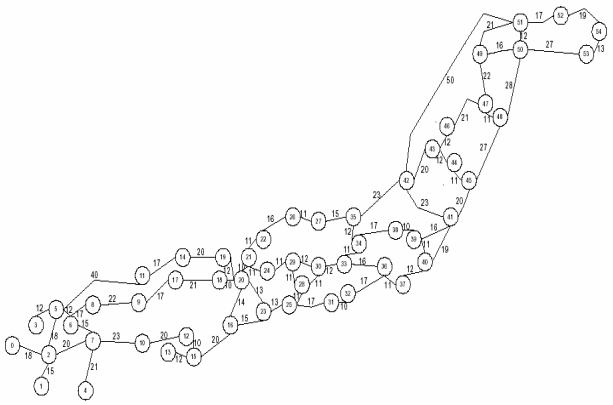


Figura 5. Topología de red NTTnet con 55 nodos y 144 enlaces.

|*U*|	*U*
10	(14,37,2) (24,5,2) (12,19,2) (9,44,2) (13,37,2) (7,48,1) (5,46,1) (2,37,1) (8,41,0) (2,54,0)
20	(17,54,2) (23,51,2) (21,0,2) (5,41,2) (0,53,2) (7,33,2) (10,44,2) (11,48,2) (6,35,2) (19,48,2) (15,35,2) (24,47,2) (5,48,1) (7,34,1) (5,33,1) (10,37,1) (7,52,1) (7,52,0) (20,40,0) (20,54,0)
30	(2,14,2) (7,29,2) (10,30,2) (14,42,2) (54,30,2) (47,11,2) (47,17,2) (6,23,2) (0,54,2) (21,50,2) (5,31,1) (8,33,2) (13,48,2) (23,45,2) (10,37,2) (19,2,2) (5,28,2) (52,53,2) (5,38,2) (16,42,2) (11,1,2) (4,51,2) (6,35,1) (2,54,1) (9,34,1)(24,51,1) (52,20,1) (17,36,1) (7,41,0) (15,46,0)
40	(2,14,2) (7,29,2) (10,30,2) (14,42,2) (47,11,2) (6,23,2) (0,54,2) (13,48,2) (8,33,2) (23,45,2) (10,37,2) (19,2,2) (5,28,2) (52,53,2) (5,38,2) (16,42,2) (11,1,2) (4,51,2) (14,37,2) (12,19,2) (9,44,2) (13,37,2) (7,48,1) (5,46,1) (2,37,1) (6,35,1) (2,54,1) (9,34,1) (24,51,1) (5,31,1) (52,20,1) (17,36,1) (7,41,0) (15,46,0) (8,41,0) (2,54,0) (24,5,0) (21,50,0) (47,17,0) (54,30,0)

Tabla 3. Conjunto de demandas unicast *U* a ser comodadas en la red óptica NTTnet.

Los resultados correspondientes a los conjuntos unicast de la Tabla 3 son presentados en la figura 6 (a-f). Puede notarse que para conjuntos de solicitudes unicast con pocas solicitudes, el SPEA continua teniendo un desempeño notable con respecto al NSGA-II. Se puede observar un desempeño similar al del problema 1, donde el SPEA es mejor en situaciones donde la carga es baja, no así cuando se le exige con mayores solicitudes y generaciones a los algoritmos, demostrándose un mejoramiento notable en estos casos del algoritmo NSGA-II.

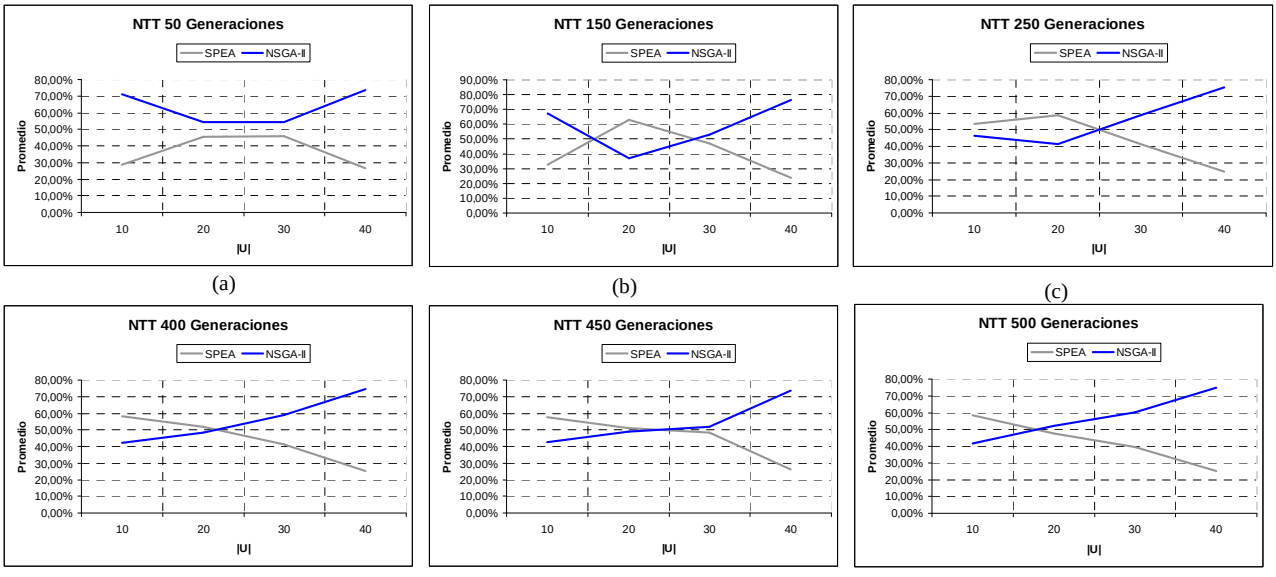


Figura 6. Resultados para el problema de prueba 2, sobre la topología de red NTTnet, correspondiente a los conjuntos de solicitudes unicast |*U*|={10,20,30,40} agrupados en generaciones de a 50 hasta las 500, definidas como parámetro.

En los experimentos con conjuntos *U* mayores, el desempeño del NSGA-II es notablemente superior al del SPEA en todas las generaciones. Para el conjunto |*U*|=30, el NSGA-II logró un 60%

por sobre los 40% del SPEA (Figura 5). Para $|U|$=40, se verifica que el NSGA-II obtuvo un 70% de las soluciones mientras que el SPEA solo obtuvo el 25% teniendo en cuenta las 500 generaciones.
Como primeras conclusiones se pude acotar que el desempeño del SPEA se va degradando conforme aumenta el número de solicitudes unicast, en las pruebas realizadas. Por otra parte, el NSGA-II tiene un desempeño complementario al SPEA. En efecto, el NSGA-II obtiene mejores rendimientos frente al SPEA a medida que aumenta la complejidad, como puede verificarse en las Figuras anteriores. Puede observarse que la curva formada para el SPEA se va degradando en su desempeño mientras que con el NSGA-II ocurre exactamente lo contrario. Se puede concluir entonces de estas pruebas que para escenarios de baja complejidad es recomendable la utilización del SPEA, mientras que el NSGA-II es muy promisorio para escenarios de mayor complejidad.

7. CONCLUSIÓN Y TRABAJOS FUTUROS

Este trabajo presenta un enfoque multiobjetivo basado en algoritmos evolutivos (MOEAs) para la resolución del problema de asignación óptima de *lightpath* bajo requerimientos de QoS. Tres niveles de QoS fueron definidos: Oro, Plata y Bronce. Para Oro, un *lightpath* alternativo y exclusivo al primario es asignado conforme el esquema de protección dedicada. En Plata, el *lightpath* alternativo asignado puede ser utilizado por otra solicitud de menor calidad, respondiendo al esquema de protección compartida. Para Bronce, no se asignan caminos alternativos, por lo que ante una eventual falla en el *lightpath* correspondiente, la transmisión de datos no será recuperada.
El SPEA y el NSGA-II fueron los enfoques utilizados para acomodar un conjunto de demandas unicast, donde se busca minimizar simultáneamente el número de saltos y número de conmutaciones de longitudes de onda, bajo restricciones impuestas por los múltiples niveles de QoS. Así, un conjunto de soluciones de compromiso llamado Conjunto Pareto es calculado en una sola corrida. Este tipo de optimización tiene una especial ventaja sobre otros métodos, como el de suma ponderada u orden lexicográfico, debido a que no necesita restricciones o conocimientos *a priori*, evitando eliminar buenas soluciones.
Resultados experimentales sobre topologías de redes conocidas como NSF y la NTT con diferentes conjunto de demandas, muestran que bajo ciertas condiciones el SPEA presenta un mejor desempeño sobre el NSGA-II. A medida que la complejidad del escenario de prueba aumenta, el NSGA-II presenta un claro aumento en su desempeño. Se puede concluir en consecuencia que no existe un enfoque claramente superior para todos los escenarios de pruebas. La utilización del SPEA es recomendable para conjuntos pequeños de demandas unicast mientras que el NSGA-II se recomienda para escenarios de mayor complejidad.
Como trabajo futuro los autores proponen tratar el problema en cuestión con otros algoritmos multiobjetivos, utilizando otras funciones objetivos como el número de transmisores/receptores ópticos, considerando otros niveles de calidad de servicios a ser propuestos.

REFERENCIAS

[1] A. M. Hamad y A.Kamal. “A survey of Multicasting Protocols for Broadcast-and-Select Single-Hop Networks”. *Network, IEEE*. Volumen 16, Edición 4, Julio-Agosto 2002, Páginas: 36 – 48.
[2] A.K. Somani . *Survivability and Traffic Grooming in WDM Optical Networks*. Cambridge University Press, 2005. ISBN-I3 978-05II-I3754-9 eBook (NetLibrary).
[3] E. Modiano y A. Narula-Tam. “Survivable routing of logical topologies in WDM Networks”, INFOCOM 2001. Proceedings. IEEE, Volumen. 1, 2001, Páginas: 348-357.

[4] S. Zhong y A. Jackel. "Optimal priority-based lightpath allocation for survivable WDM networks", ICCCN 2004. Proceedings. IEEE, 2004, Páginas: 17-22.
[5] S. Ramamurthy y B. Mukherjee. "Survivable WDM Mesh Networks, Part I-Protection", Proceedings. IEEE INFOCOM'99, Páginas: 744-751, Marzo, 1999.
[6] G. Ellinas, A.G. Hailemariam y T.E. Stern, "Protection cycles in Mesh WDM Networks", *IEEE Journal on Selected Areas in Communications*, Volumen 18, Nº 10, Páginas: 1924-1937, Octubre 2000.
[7] D. Stamatelakis y W.D Grover. "IP Layer Restoration and Network Planning Based on Virtual Protection Cycles", IEEE JSAC, Volumen 18, Nº 10, Páginas: 1938-1949, Octubre 2000.
[8] S. Ramamurthy y B. Mukherjee, "Survivable WDM Mesh Networks-II: Restoration". ICC'99. 1999 *IEEE International Conference on Communications*, Volumen 3, Páginas: 2023-2030, 1999.
[9] R. R. Irashko, y W. D. Grover, "A highly Efficient Path-Restoration Protocol for Management of Optical Network Transport Integrity". *IEEE Selected Areas in Communications*, Volumen 18, Nº 5, Mayo 2000, Páginas: 779-794.
[10] K. Miettien. "Some methods for nonlinear multi-objetive optimization". En E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, y D. Corne, editores, *First International Conference on Evolutionary Multi-Criterion Optimization*. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
[11] A. S. Tanenbaum, *Computer Networks*, Prentice Hall, Cuarta Edición, 2003.
[12] D. A. V. Veldhuizen y G. B. Lamont. "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-
Art". *Evolutionary Computation*. Páginas: 125-147, 2000.
[13] D. Pinto y B. Barán. "Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimization approach". *2nd IFIP/ACM Latin American Networking Conference 200*5 - LANC'05. Cali - Colombia. Octubre 2005.
[14] D. B. Fogel, editor Evolutionary Computation. "The Fossil Record Selected Reading on the History of Evolutionary Algorithms". The institute of Electrical and Electric Engineers, New York.
[15] D. Van Veldhuizen, "Multiobjective Evolutionary Algorithms: Classifications, Analysis, and New Innovations", PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Mayo 1999.
[16] A. Dias y J. De Vasconcelos. "Multiobjective Genetic Algorithms Applied to Solve Optimization Problems". *IEE Transactions on Magnetics*, Volumen 38, Nº 2, March 2002.
[18] E. Zitzler, y L. Thiele, "Multiobjective Evolutionary Algorithms: A comparative Case Study and the Strength Pareto Approach", *IEEE Trans. Evolutionary Computation*, Volumen 3, Nº 4, 1999, Páginas: 257-271.
[19] K. Deb y T. Goel. "A Fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Trans. Evolutionary Computation*, 6(2): 182-197, 2002.
[20] M. Chakraborty y U.K. Chakraborty. "An analysis of linear ranking and binary tournament selection in genetic algorithms". *IEEE Communications Society. Information, Communications and Signal Processing*, 1997. ICICS., Proceedings of 1997. Volume 1, Edición, 9-12 Septiembre 1997, Páginas: 407 – 411.
[21] I. Chlamtac, A. Ganz y G. Karni. "Lightpath Communications: An Approach to High Bandwidth Optical WANs". *IEEE Transactions on Communications*. Volumen 40, Nª 7, Páginas: 1171- 1182, July, 1992.
[22] F. Talavera, J. Prieto, J. Crichigno y B. Barán. "Comparación de Algoritmos Evolutivos Multi-Objetivos". *Congreso Argentino de Ciencias de la Computación - CACIC'2004*. Bs. Aires

Using JOP to build a chip multiprocessor JVM for embedded realtime systems

José Pablo A. Andreotti.
Dpto. de Arquitectura de Computadoras, Universidad Nacional de Córdoba.
Córdoba, 5000, Argentina.
albertoandreotti@gmail.com

Abstract

Programming embedded devices has been historically a difficult task, involving the learning of the features of a specific device such as the assembly language of a processor. This led to the use of Java in embedded system as a way to improve the development process in such systems.
The challenge of using Java in embedded systems was targeted in part, by the research effort of the JOP (Java Optimized Processor) project carried out by Martin Schöeberl [1], at TU Vienna. JOP is a small processor specifically designed for the execution of real-time Java programs.
This paper discusses the possibilities and the challenges that arise when I built a CMP (chip multiprocessor) JVM (Java Virtual Machine) based on JOP as part of my thesis to obtain my degree in computer engineering. It describes which hardware modules were necessary to be added in order to achieve shared memory access and a proper synchronization of the processors. Then, the problem of scheduling multiple threads on multiple processors and meeting the requirements imposed by the JMM (Java Memory Model) is discussed. All the proposed solutions use a simplistic approach and serve as a baseline for further research.
Finally, a brief description of an implementation of the system in a FPGA (field programmable gate array) is given.

Keywords: JVM, JOP, CMP, FPGA.

1. Introduction

Our intention is to explore the design space for a multiprocessor system that meets the following characteristics:

a) Support for a scalable multiprocessor embedded JVM.
b) Support for multiple threads running on top of each processor.
c) The gain in performance is achieved with little impact on the programmer.
d) The sintaxis and semantics of the Java language remain unchanged.

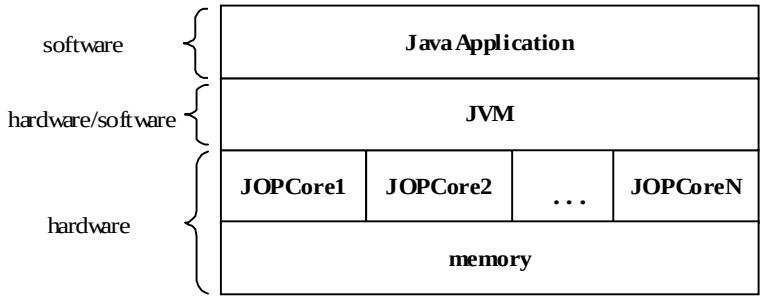


Figure 1.1 Organization of a system to meet the above mentioned requirements. The Java application lies on a JVM which is implemented in both hardware and software. To provide its functionalities the JVM relies on several processors that access a shared memory system.

A layered diagram showing a configuration for the desired system can be seen in figure 1.1.
In order to obtain a system like this the following problems need to be addressed:

a) Provide a way for the processors to share a common memory system.
b) Find a way to make communication available among the processors, so they can cooperate in performing tasks such as initialization and synchronization.
c) Provide a scheduler capable of assigning threads to several processors for their execution.
d) Verify that the system does not violate the restrictions imposed by the Java Memory Model.

The rest of this paper explains how to tackle these four topics.

2. Provide a way for the processors to share a common memory system

In order to implement a shared memory architecture, we need to provide shared access to a single memory interface. In JOP, the interconnection between the processor, the memory interface and the I/O devices is achieved using a standard interface called SimpCon [2].
SimpCon provides a way of accessing different kind of devices in a seamless manner. The SimpCon standard makes a division between slave and master devices. It also contemplates the possibility of multiple access to a single slave device through an arbiter device.
The arbiter performs two basic tasks:

1. Performs transactions on the shared memory system on behalf of the masters(i.e., the JOP cores).
2. Implements the resource sharing policy.

The resultant memory architecture is depicted in figure 2.1.

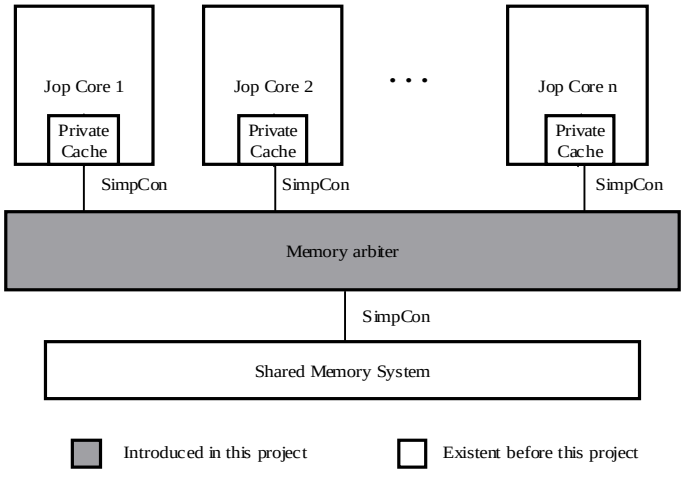


Figure 2.1 Resulting memory architecture after introducing the memory arbiter. The JOP cores are master devices and the shared memory interface is a slave device.

In this configuration, the memory arbiter acts as a master device to the memory interface and as a slave to the JOP cores. It reproduces the transactions started by the masters on the memory interface. The arbiter is transparent to the JOP cores, the only effect noticed by the cores is an eventual increase in memory latency cycles due to the time sharing mechanism.

The resource sharing policy implemented in this project is the simple round robbin. Further research might include a comparison of performance achieved with different resource sharing policies against different benchmark applications.

3. Find a way to make communication available among the processors, so they can cooperate in performing tasks such as initialization and synchronization.

To achieve proper initialization of the CMP JVM and to give support to Java monitors we need a means of communication among the processors. The proposed solution consists in introducing an extra I/O device which implements a common shared control bus. The resulting configuration is shown in figure 3.1.

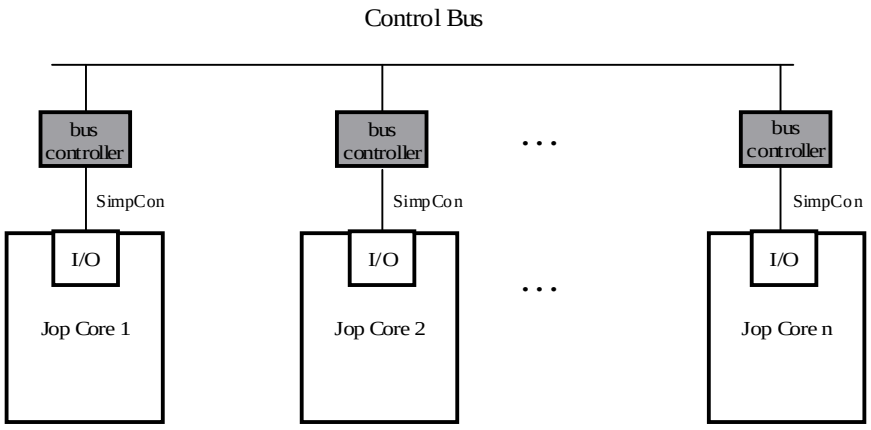


Figure 3.1 The control bus. The bus controller I/O device is used to provide access to a common control bus shared among all the procesors.

The control bus provides two functions:

Initialization: To properly initialize internal structures of the JVM such as internal memory addresses and the GC(Garbage Collector) we need a means of communication between processors. The alternative chosen in this project was to let one processor be the Master[1] and perform all the initializations. When the initialization is done, the master processor signals the other slave processors using a dedicated signal on the control bus. A sequence diagram showing the whole process is shown in figure 3.2.

Synchronization: Synchronization among threads in Java is performed through monitors. As stated in [3] to achieve mutual exclusion in O(1) time in a multiprocessor environment, we need to perform some kind of atomic access to the memory subsystem.
In order to provide such an atomic access we introduced memory locks which are implemented through the control bus as depicted in figure 3.3.

[1] The word "Master" here is no used in the context of the SimpCon standard.

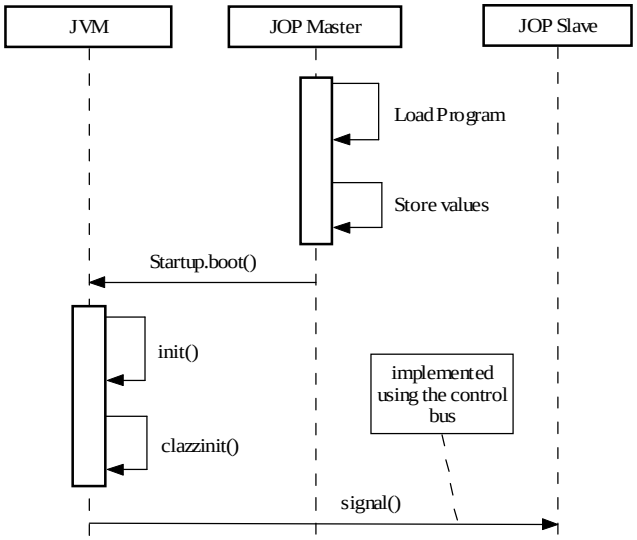


Figure 3.2 Initialization of the system. The master processor performs initializations such as loading the program into main memory, storing values, initializing the GC, etc. When all initialization is done, the master signals the other processor(s).

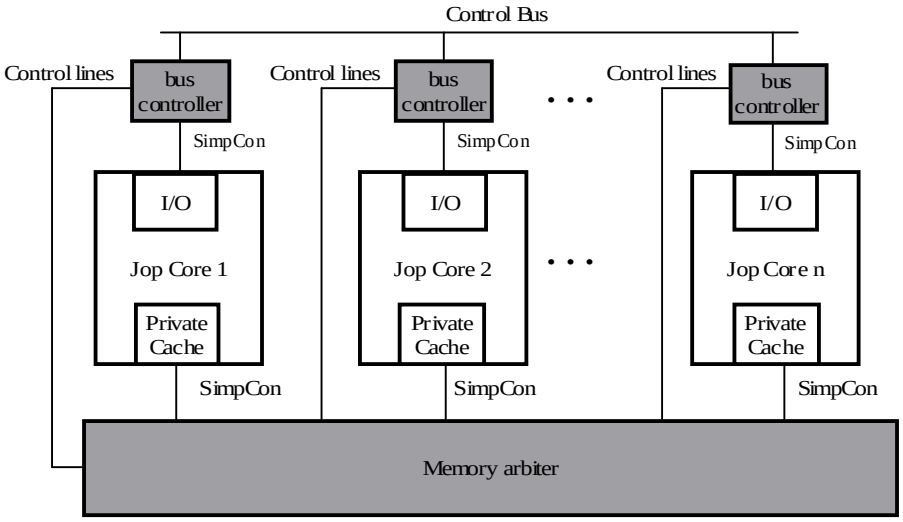


Figura 3.3 Interconnection of the bus controller to the Memory Arbiter. The lines labelled "Control lines" are used to negotiate atomic access to memory through a simple protocol. The responsible of granting atomic access to memory is the memory arbiter.

Using memory locks, the time for a thread to enter its critical section is constant. We only use memory locks inside the code of the JVM. We use them both to impose and order on the accesses to certain internal structures of the JVM and in the implementation of Java monitors.
For example, we have the following pseudocode[1] for the implementation of *monitorenter* and *monitorexit*[2],

monitorenter:
1. Request atomic memory access.
2. Read the owner_id field (identity of the owner of the object).

monitorexit:
1. Load the cnt field.
2. Decrease the cnt field.
3. Store it back to memory.

[1] Bytecodes can be implemented in hardware, in Java or in the microcode native to JOP.
[2] Monitorenter and monitorexit are the bytecodes used to implement synchronized blocks in Java.

3. If the owner is that thread, release memory and jump to 6.
4. If the owner is other thread, release memory and jump back to 1.
5. If the object is not owned by any thread, put this thread id as the new owner, release memory.
6. Increase cnt (a counter which represents the number of times the monitor was entered)
7. Fetch the next bytecode.

4. If cnt=0, release the monitor. This is accomplished by setting owner_id to zero.
5. Fetch the next bytecode.

Further research might contemplate the relative performance of different algorithms and the potential advantage of using lock avoidance mechanisms like the ones described in [4].

4. Provide a scheduler capable of assigning threads to several processors for their execution.

4.1 Introduction

The problem of handling multiple flows of instructions in Java is contemplated at the language level. As we can find in The Java™ Language Specification *Third Edition* [5], Java threads are the means by which Java give support to concurrent programs. The specification also makes no difference on whether Java threads are implemented on one or several multiprogrammed processors.
A task switch in a real-time system consists of two parts: *scheduling* and *dispatching*. Scheduling involves the selection of the next task to be executed and dispatching involves the actual context switch of the processor. So, given a set of threads, the design space is determined by the decision of how to distribute these tasks among the processors.

4.2 Single centralized Scheduler vs several Schedulers

As we stated above for each thread we can:

1. Let just one processor perform its scheduling and dispatching (i.e., “tie” the thread to a processor).
2. Let any processor perform the scheduling and dispatching of a thread.

The second option above is an attractive alternative because it allows the inclusion of interesting features in the scheduler such as allowing one thread to switch among processors. Implementing a centralized scheduler involves some modifications in the way JOP handles interrupts. That’s because the JOP scheduler is very dependent on the interrupt subsystem of each processor. So, in order to implement a centralized scheduler we need a centralized interrupt subsystem shared among all the processors.
On the other hand, the first option is simpler and involves no modification of the scheduler or the interrupt subsystem. This is the one that was implemented for that project. We spend the rest of this section describing how this option was implemented.
The overall architecture for a dualcore JVM is depicted in figure 4.1. In this configuration, the user

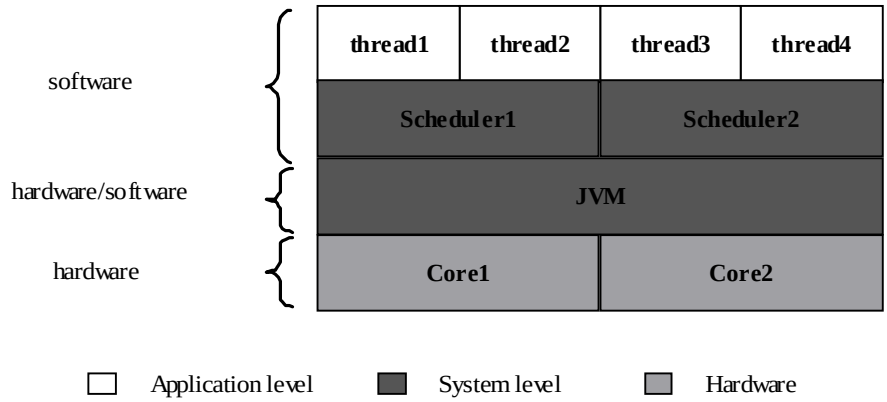


Figure 4.1 Overall architecture of a dual core JVM with two independent schedulers. Each application is written as a set of threads. Each thread is then "tied" to a specific scheduler which runs on one processor. JVM code is accessed by scheduler 1 and scheduler 2 and is thus executed on both cores.

chooses the processor each thread will run on at the time of the creation of the thread objects. This object creation process is shown on figure 4.2.

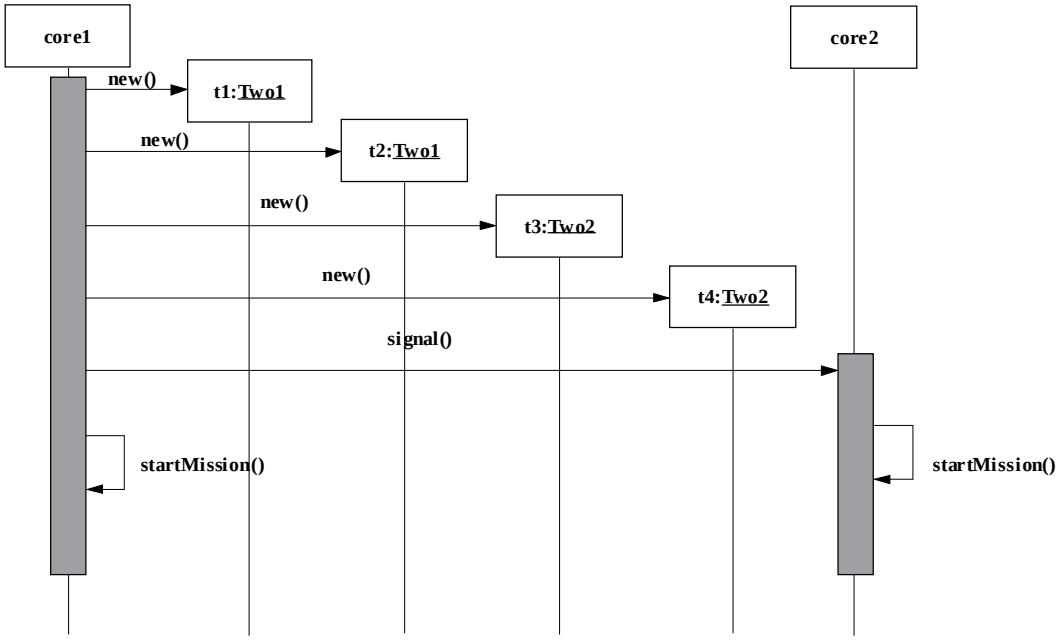


Figure 4.2 Creation of thread objects. Objects of the classes Two1 and Two2 represent threads that will be scheduled on the first and second cores respectively. In this application Core 1 creates the objects and then "signals" the other core. Core 2 starts scheduling its threads with the call to startMission().

An obvious disadvantage of this approach is that the user must be aware of the existence of different processors. On the other hand, it might result attractive in some real-time applications where it is necessary to ensure the availability of certain resources for a specific task. In such a case we just need to "tie" the thread representing the task to the processor that has access to the resources. This is accomplished by creating an appropriate thread object for that task.

5. Verify that the system does not violate the restrictions imposed by the Java Memory Model

As stated in [6], the access to memory in a multiprocessor system is ordered by a shared memory consistency model. The JMM is an abstract model that defines the allowed behaviours of a multithreaded program.
In a regular system, we have two memory models: the memory model of the underlying architecture and the JMM. This is shown is figure 5.1. The JMM defines which are the valid transformations when a compiler produces bytecode, when the JVM produces native code and when the hardware applies optimizations on the native code.

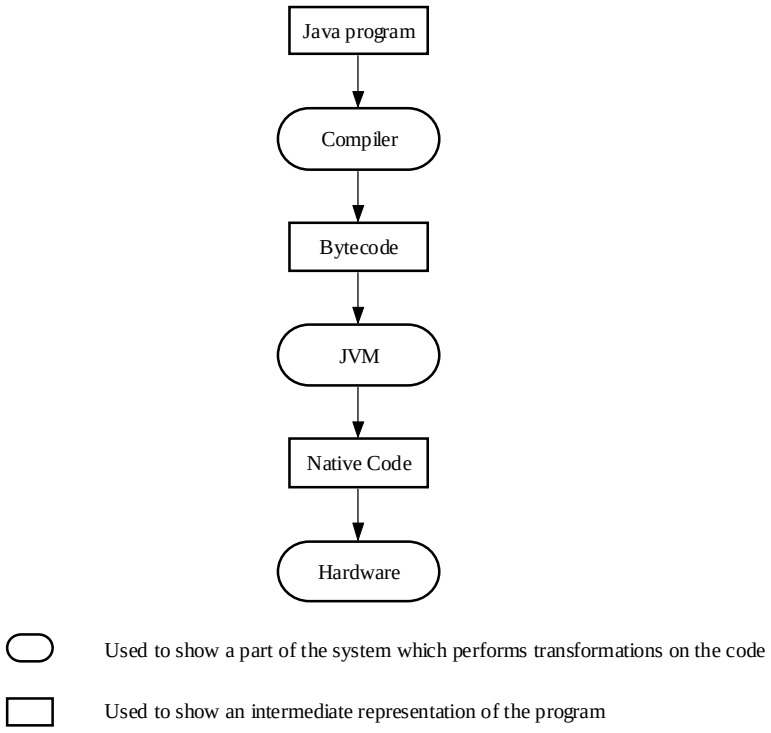


Figura 5.1 Different transformations and intermediate representations for a Java program. In a typical Java system a program is first compiled to a standard bytecode representation, then the JVM generates object code native to the underlying processor. Finally the hardware might perform some transformations on the resulting instruction flow such as instruction scheduling.

Many restrictions of the JMM arise when running a multithreaded program on processors that perform *instruction reordering.* Instruction reordering is a technique used to increase the instruction throughput in a processor pipeline. The idea is to rearrange instructions to get rid of data dependences among them and reduce the number of stalls in the processor pipeline. These data dependences are known as *data hazards*[1].
As demonstrated in chapter 5 of JOP: A Java Optimized Processor for Embedded Real-Time Systems, [1] data hazards are not present in the processor pipeline of JOP. So, when building a CMP JVM using JOP it is not necessary to consider JMM issues related with instruction reordering.
Another issue commonly found in multiprocessors is *cache coherence*[2]. Due to the memory hierarchy in JOP, no problems arise related to cache coherence when interconnecting several

[1] For an in depth study of data hazards, see Patterson & Henessy, Computer Architecture a quantitative approach, [7].
[2] See Parallel Computer Architecture, David E. Culler and Jaswinder Pal Singh, [8].

processors together. In JOP, only data local to a thread is cached[1]. This data is never shared among threads, so no coherence problems are possible . Class variables are stored in main memory. At any moment only a single copy of these variables exists. As there is no data duplication data coherence problems are not possible.

6. Description of an implementation of the system in a FPGA

Both JOP and all the hardware devices introduced in this project are implemented using the VHDL(VHSIC hardware description language)hardware description language. Devices described in VHDL can be implemented in a FPGA. In this section a description of an implementation of the system in a FPGA is provided.

The current implementation consists of a dual core JVM in an FPGA provided by Altera. Devices found in a typical JOP configuration such as the serial interface and external I/O ports were distributed among the processors. Each processor has its own timer and bus controller modules.
A resource utilization comparison of the system against a single core JVM is provided in table 6.1.

	Logic Cells	Memory	Utilization
Single Core JVM	2661	35840 bits	100%
Core	1094		41.11%
Extension	205		7.70%
Scio	316		11.87%
Mem	968		36.37%
Rest of the system	78		2.93%
Dual Core JVM	5409	71680 bits	100 %
Memory arbiter	420		7.76 %
Bus controller	15		0.27 %
Rest of the system	4974		91.97 %

Table 6.1 FGPA utilization for both the single core and dual core versions of the JVM using JOP. The single core version has a 4Kb cache while the dual core version has a 2Kb cache for each core

.

We can see that the device utilization for the bus controller and memory arbiter is 8.3%. So that, we can conclude that little amount of resources are necessary to perform the interconnection of a dual core JOP based JVM.

References

[1] Martin Schöberl. JOP: A Java Optimized Processor for Embedded Real-Time Systems.
[2] Martin Schöberl. SimpCon – a Simple SoC Interconnect.
[3] Michael L. Scott. Programming Language Pragmatics.
[4] Kiyokuni Kawachiya, Akira Koseki, Tamiya Onodera. Lock Reservation: Java locks can mostly do without atomic operations.
[5] Tim Lindholm, Frank Yellin. The JavaTM Virtual Machine Specification 2nd Edition.
[6] Sarita V. Adve, Kourosh Gharachorloo. Shared Memory Consistency Models: A Tutorial.

[1] The local data cache in JOP is called the *stack cache*, see [5].

[7] John Hennessy and David Patterson. Computer Architecture: A quantitative approach 3rd edition.
[8] David E. Culler y Jaswinder Pal Singh, Parallel Computer Architecture.

A Semi-Automatic Method for Ontology Mapping ⋆

Laura Perez, Agustina Buccella, and Alejandra Cechich

GIISCO Research Group
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue
Neuquen, Argentina

Abstract. Ontology mapping involves the task of finding similarities among overlapping sources by using ontologies. In a Federated System in which distributed, autonomous and heterogeneous information sources must be integrated, ontologies have emerged as tools to solve semantic heterogeneity problems. In this paper we propose a three-level approach that provides a semi-automatic method to ontology mapping. It performs some tasks automatically and guides the user in performing other tasks for which his/her intervention is required. Finally, a plug-in of the ontology editor, Protégé, is presented showing how the method is implemented through a case study.

1 Introduction

A Federated System [7] refers to the data integration of distributed, autonomous and heterogeneous information sources. In general it is implemented by using a 4-layer architecture [3], in which the *federation layer* is the core of the system. In this layer, each information source is described by its own ontology (domain ontologies), and all of them converge in one global shared vocabulary. This shared vocabulary contains basic terms (the primitives) of the whole domain. In this way, an hybrid ontology approach [18] is followed.

Several proposals on ontology mapping have emerged in the last years [9]. Among them, we can cite [13, 12, 11, 14]. For example, [13] and [12] propose two similar ontology-merging tools. On one hand, the PROMPT tool described in [13] proposes an interactive tool that guides the user through the merging process. However the main problem with the PROMPT tool is that it is highly dependent on the names of the concepts in the ontology. On the other hand, Chimarea [12] provides support for merging of ontological terms from different sources, checking the coverage and correctness of ontologies and maintaining ontologies over time. Except for several situations referring to structural aspects of the ontologies, Chimarea does not make any suggestion to the user; and the only relation that Chimarea considers is the subclass/superclass relation. Another proposal for semantic matching is introduced in [11], where a lexical and a conceptual layer are used to find similarities. At the lexical level, the method uses a lexical function called lexical similarity measure (SM). At the conceptual level, concepts (classes and properties) are compared taking into account the taxonomies in which they appear. However, some types of properties are not considered by this method. Finally, the proposal of Rodriguez et.al [14] presents a combination of two different approaches to similarity assessment – the feature matching process [17] and the semantic distance. Common features increase the similarity value and distinct features decrease it. The main disadvantage with this method is that the similarity values cannot be calculated neither automatically nor semi-automatically due to the high dependence on natural language descriptions.

In previous work [5, 4], we have proposed a three-level approach that allows us to build similarities expressed as mappings. In this work we improve this method taking into account

⋆ This work is partially supported by the UNComa project 04/E059 (Mejora del Proceso de Desarrollo de Software Basado en Componentes).

cycles in the ontologies and implementing a tool as a plug-in of the ontology editor Protégé [1]. This extension allows us to introduce a semi-automatic process for ontology mapping.

This paper is organized as follows: Section 2 shows the main steps of our three level approach describing changes performed to detect and solve cyclic ontologies. Then, in Section 3 the structure of components for our plug-in is described. An example describing how our tool works is shown in Section 4. We discuss future work and conclusions afterwards.

2 A Three-Level Approach to Ontology Mapping

In previous work, we have proposed a three-level method to ontology merging, taken into account information ontologies provide [5]. In this way, concepts of an ontology are compared using three comparison levels: *syntactic*, *semantic* and *user* level. Figure 1 shows our approach graphically, where the levels are part of the process.

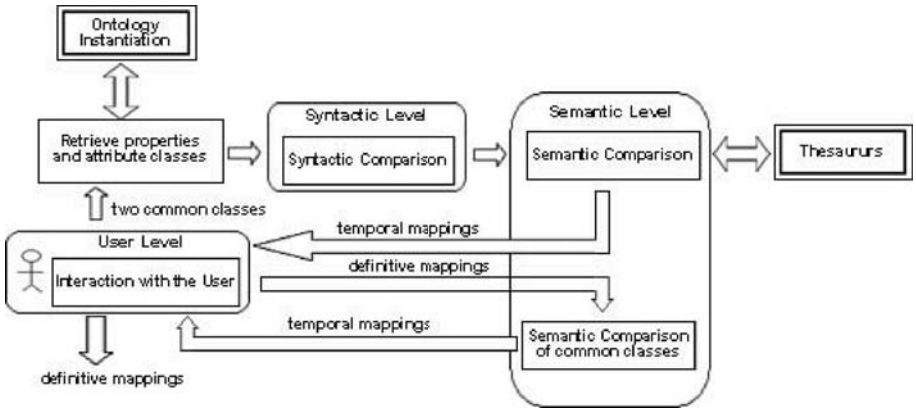


Fig. 1. Approach for searching similarities

Ontology Instantiation and *Thesaurus* are the two external modules. The former, obtains the object structure from ontologies described in some ontology language. Figure 2 shows how the different elements of an ontology are divided. The first division refers to two different elements. On one branch we have the *classes* and on the other branch the *properties*. Firstly we analyze the *classes* branch, which is also divided into two new branches: *common classes* and *attribute classes*. Both are classes defined in the ontology to represent things about the world. The specific role defined in the ontology is the difference between them. The *common classes* have the role of representing things about the domain and the *attribute classes* have the role of representing information about a common class. Both roles exist because some concepts of the ontologies act as attributes. For example, an ontology can have the *Animal* class as a common class and the *Organ* class as an attribute class because *Organ* exists to describe a characteristic about a common class. The *Organ* class has no properties.

On the other branch, Figure 2 shows the *properties* branch which is also divided into two new branches: *datatype properties* and *special properties*. A property is a set of tuples that represents a relationship among objects in the universe of discourse. Each tuple is a finite, ordered sequence (i.e., list) of objects. The properties have restrictions to denote functions, cardinality, domain, range, etc. The *datatype properties* are properties relating a class or a set of classes with a data type. For example, the animal name is a common property between the

Animal class and the *String* data type. On the other hand, the *special properties* are properties relating classes. For example, the relationship between the *Animal* class and the *Organ* class to denote the organs of an animal. Thus, a common class has both datatype properties and special properties, and attribute classes do not have properties.

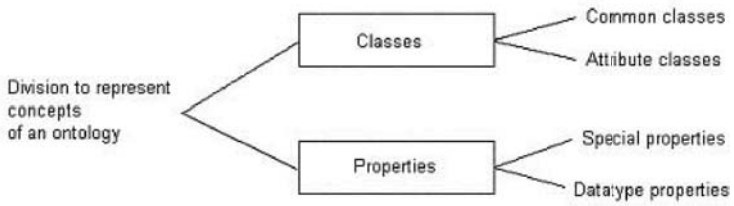


Fig. 2. Proposed division to represent the ontology

The latter, the *Thesaurus* module, uses thesauruses to search for synonyms, which are detected by the module through the use of a similarity function.

In Figure 1, two common classes (of different ontologies) are indicated by the user. These two classes are inputs of the *Retrieve Properties and Attribute Classes* module, which retrieves the attribute classes and special and datatype properties of each class by using the object structure of each ontology. This retrieved information enters the *Syntactic Comparison* module, which analyses syntactically classes and properties relating with the concepts. A set of syntactic functions [5] are used here.

Then, the *Semantic Comparison* module compares the classes and properties semantically. To do so, we extract semantic information from the Thesaurus module in order to find synonym relationships. Using the results of the syntactic level functions, we construct functions that combine these values together with the thesaurus information.

In the *Interaction with the User* module all the mappings that exceed a threshold are shown to the user, and he decides if the mappings are correct. The accepted mappings are classified as definitive mappings.

The *Semantic Comparison for Common Classes* module receives the definitive mappings and compares the common classes of the two ontologies. It uses the mappings added by the comparison of properties in order to denote the set of similar attributes (properties) of both classes.

One more time, in the *Interaction with the User* module, all mappings are displayed to the user and he decides if these mappings must be added permanently.

2.1 Improving the Method: Detecting Cycles

Our approach analyzes the ontologies as graphs, taking into consideration both taxonomic and non taxonomic relationships among terms.

Figure 3 describes our basic method for searching similarities. The method has a series of steps depending on the different elements of the ontology (see Figure 2).

Firstly, a user must indicate the first mapping, for example between the *Animal* class of one ontology and the *Creature* class of the other ontology. If the classes are common classes, the system compares firstly the datatype properties of both classes. The *cleaning_process* in the method

Similarity(O1,O2)

the user enter to similar classes (c1,c2)
if (c1 and c2 are common classes)
for each datatype property $dtp_i \in c1$ and $dtp_j \in c2$
$cleanning\_process(dtp_i, dtp_j)$
$sim1_{thesaurus}(dtp_i, dtp_j) = search\_on\_thesaurus(dtp_i, dtp_j)$
$sim1_{sint}(dtp_i, dtp_j) = w_{ed} * sim_{ed}(dtp_i, dtp_j) + w_{tri}*$
$sim_{tri}(dtp_i, dtp_j) + w_{dtc} * sim_{dtc}(range\_of(dtp_i), range\_of(dtp_j))+$
$w_{thesaurus} * sim1_{thesaurus}(dtp_i, dtp_j)$
if $sim1_{sint}(dtp_i, dtp_j) \geq th_{accept}$
$add\_mapping(dtp_i, dtp_j)$
for each special property sp_i $\in c1 and sp_j \in c2$
$cleanning\_process(sp_i, sp_j)$
$sim2_{thesaurus}(sp_i, sp_j) = search\_on\_thesaurus(sp_i, sp_j)$
$sim_{rest}(sp_i, sp_j) = check\_restrictions(sp_i, sp_j)$
$sim2_{sint}(sp_i, sp_j) = w_{ed} * sim_{ed}(sp_i, sp_j) + w_{tri} * sim_{tri}(sp_i, sp_j)+$
$w_{thesaurus} * sim2_{thesaurus}(sp_i, sp_j) + w_{rest} * sim_{rest}(sp_i, sp_j)$
$if(c1, c2) \notin Visited$
add_visited(c1,c2)
if $(range\_of(sp_i), range\_of(sp_j)) \notin Mapped$
$sim_{total}(sp_i, sp_j) =$
$calculate\_all\_the\_process\_for(range\_of(sp_i), range\_of(sp_j))$
if $(range\_of(sp_i), range\_of(sp_j)) \in Mapped$
$sim_{total}(sp_i, sp_j) = get\_value(range\_of(sp_i), range\_of(sp_j))$
$sim_{sp}(sp_i, sp_j) = w_{sint} * sim2_{sint}(sp_i, sp_j) + w_{total} * sim_{total}(sp_i, sp_j)$
if $sim_{sp}(sp_i, sp_j) \geq th_{accept}$
$add\_mapping(sp_i, sp_j)$
remove_visited(c1,c2)
using the added mappings
$cleanning\_process(c1, c2)$
$sim3_{thesaurus}(c1, c2) = search\_on\_thesaurus(c1, c2)$
$sim3_{sint}(c1, c2) = w_{ed} * sim_{ed}(c1, c2) + w_{tri} * sim_{tri}(c1, c2)+$
$w_{thesaurus} * sim3_{thesaurus}(c1, c2)$
if c1 and c2 are attribute classes
$sim_{class}(c1, c2) = sim3_{sint}(c1, c2)$
if c1 and c2 are common classes
$sim_{class}(c1, c2) = w_{sint} * sim3_{sint}(c1, c2) + w_{propiedadtes\_especiales}$
$*sim_{propiedadtes\_especiales} + w_{propiedadtes\_tipo\_de\_dato} * sim_{propiedadtes\_tipo\_de\_dato}$
$if(c1, c2) \notin Visited$
if $sim_{class}(c1, c2) \geq th_{accept}$
$add\_mapping(c1, c2)$
$if(c1, c2) \in Visited$
if $sim_{class}(c1, c2) \geq (th_{accept} - (th_{accept} * w_{propiedadtes\_especiales}))$
$add\_mapping(c1, c2)$

Fig. 3. Steps for searching similarities

denotes the process of elimination of articles, prepositions and non-relevant characters (_,:,-, etc.). Next, thesauruses are used to search for synonymies. The function $sim1_{thesaurus}(dtp_i, dtp_j)$ is equal to 1 if a synonym relationship is found for the two datatype properties and it is equal to 0 otherwise.

Then, the $sim1_{sint}(dtp_i, dtp_j)$ function is calculated using four syntactic functions. The *edit distance* function, which considers the number of changes that must be done to turn one string into the other, and weights the number of these changes with respect to the length of the shortest string. The *trigram* function [10], which is based on the number of different trigrams in two concepts or strings. The *data type compatibility* function, $(sim_{dtc}(range\_of(dtp_i), range\_of(dtp_j)))$ which compares the datatype of the ranges. And the result of applying the *thesaurus* function, aforementioned. The $sim1_{sint}(dtp_i, dtp_j)$ function returns a value between 0 and 1; and the sum of weights, the w values (w_{ed}, w_{tri}, w_{dt} and $w_{thesaurus}$), is equal to 1.

Finally, if the result of the function exceeds a threshold (th_{accept}), a temporal mapping is added.

The $sim_{rest}(sp_i, sp_j)$ function checks special property restrictions [16] such as functional, symmetric, allValuesFrom, someValuesFrom, cardinality, etc. That is, it compares the constrains applied to the properties. Only when both properties have the same restrictions, the function returns 1; otherwise it returns a percentage according to the number of restrictions that are the same.

Then, a temporal mapping is added when the $sim_{sp}(sp_i, sp_j)$ function exceeds the threshold.

Following, the method compares the special properties included in the common classes. The comparison is similar to the previous case, but the *datatype compatibility* function is not calculated.

The $sim_{total}(sp_i, sp_j)$ function makes all the similarity process taking into account the range of the special properties. Therefore, this is a recursive method that will stop when the ranges are attribute classes (because they do not have properties). Once again, thesauruses are used to find synonymies relationships.

We have detected the presence of cycles in the similarity search graph. There is a major cause for cycles, the way special properties and classes can be combined are not acyclic graphs themselves. As a result, the cycles existing in the ontology become cycles in the similarity search graph, it means, a descendant of a concept could be simultaneously an ancestor of this concept. For instance, in our similarity searching method, the similarity value of a pair of classes A and B depends on the resolution of the similarity value of a pair of classes C and D which depends on the similarity values of A and B.

Because the ontology graph contains cycles, precautions must be taken for avoiding loops in the graph traversal. It is necessary for the method to eliminate cycles. In addition, once a cycle is found with a graph's node a partial similarity value has to be calculated in order to go on with the analysis. First of all, our approach for detecting cycles is to mark as visited the nodes while covering the graph in order to avoid visiting twice the same node. This is a very simple and effective approach. So, in Figure 3 the sentence $if(c1, c2) \notin Visited$ verifies whether the node is in the analysis path. If this condition is true, then the algorithm takes places again for the range classes with the purpose of finding the sim_{total} value. On the other hand, $if(c1, c2) \in Visited$ means that the node has been visited so that no special properties will be analyzed.

Although a cycle may be detected in a node graph, the similarity search algorithm must still go ahead and return a partial value for that node. In this case the similarity value, will

be obtained only from the information available on the node itself. For instance, semantic and syntactic analysis over the concepts will be carried out but no structural analysis over the special properties will be done.

Finally, the method compares the classes. This comparison is made using the syntactic functions for common and attribute classes and the semantic function for common classes. The semantic function uses the mappings added by the property comparisons in order to denote the set of similar attributes of both classes. A temporal mapping is added if the final function exceeds the threshold. Notice that in the final similarity value for any pair of classes $sim_{class}(c1, c2)$, the operand $w_{propiedadtes\_especiales} * sim_{propiedadtes\_especiales}$ ought to be null if a cycle was detected on $(c1, c2)$. Thus, the similarity threshold is lower in this case with the intention of giving more chances to the user to decide over the mapping of this classes.

Once all similarity values are obtained for two classes, the temporal mappings are displayed to the user and he/she must decide if these mappings must be added permanently. Thus, the user makes the final decision.

One last thing to point out over the method is the check done before the process starts again over the range classes. Taking into account that either of the ontologies having a lot of classes and properties will generate a large graph, a question whether a pair classes are already in the defined mappings set is stated with the $if(range\_of(sp_i), range\_of(sp_j)) \in Mapped$ sentence. If they have been mapped, then the found value is taken to avoid doing the same analysis again.

3 Architecting a Supporting Tool

As an implementation of our three-level approach for ontology integration, we built a plug-in for the Protégé ontology editor [1]. Basically, at its core, Protégé implements a rich set of knowledge-modelling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Further, Protégé can be extended by way of a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge-based tools and applications. The Protégé platform supports two main ways of modelling ontologies through the following editors *Protégé-Frames* y *Protégé-OWL*. In this section, we will describe the design of the Protégé plug-in called OWLSim.

During the design process, the Responsibility Driven Design (RDD) [20, 19] model was used to keep our focus on the behavior of our software. This methodology helps us to identify the application's responsibilities and to divide them into collaborative objects.

The plug-in's architecture design is based on two architectural styles the model-view-controller (MVC) and its successor the presentation-application-control PAC [2, 6]. We describe how software objects are organized, this means how objects are located in components. Each component contains characteristic object roles that are located according to both the application components' functionality and the object's roles.

Figure 4 shows our plug-in's architecture. In the first place, the Transactions (*Control and business logic*) component includes the objects that are responsible for the control and business logic. Furthermore, this component mediates the interaction between Domain Model and Presentation in order to avoid direct dependency between them. Secondly, the Domain Model (*Abstraction*) component contains all those objects that represent the domain concepts. Lastly, the User interaction component (*Presentation*) is structured into objects that provide window, menu, and dialog functionality. They manage the inputs and translate them into service requirements.

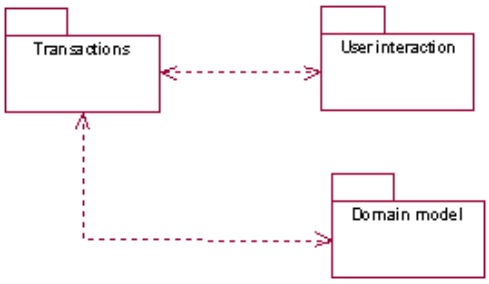


Fig. 4. Plug-in's architecture

3.1 Designing classes

One of the most important activities in Object-Oriented design is to identify object classes. Thus, we move from the requirements and descriptions – the method specification explained in previous sections (behavior that the plug-in must accomplish), the definition of OWL ontologies and the Protégé model specification – to find and describe the most important classes.

Domain and application-specific objects Domain objects represent concepts in a specific field of interest. In our domain, the ontologies and their elements, the mappings between them, and the similarity method.

Figure 5 shows part of the domain class diagram using UML notation [8] – the part where the ontology and both their elements and mappings are modelled. There are classes that model the most important components of an OWL [16] ontology, such as classes, properties and restrictions. In addition, as described in section 2, we also take into account the division of the ontology elements that the similarity method embodies (*Atributte_Class*, *Common_Class*, *Datatype_Property* and *Special_Property*). As shown in the diagram, both classes *Atributte_Class* and *Common_Class* were modelled as a class *Class* specialization. Because of their differences, the method gives different treatment to each of them. In addition, the *has_superclass* relation represents taxonomic relations in an ontology. Regarding class *Property*, it has a specialization into two classes: *Datatype_Property* and *Special_Property* in accordance to the method's division on property elements, as only *Special_Property* relates the *classes*. Moreover, both special properties and datatype properties ranges are different, so two distinct relations *has_classrange* and *has_XMLDatatyperange* were modelled to associate them to the range classes *XMLDatatype* and *Class*. Further, the *has_compatibility* association shows that each data type might be compatible with other data types. Unlike *Common classes*, that might have both types o properties, the *Attribute classes* have not gotten any property; thus, the association *has_property*is between the subclass *Common_Class* and the superclass *Property*. Finally, properties restrictions are also modelled.

Following, the mappings found by the method are contained in the *Mapping* class. The *Property_Mapping* and *Class_Mapping* classes are its subclasses. The former class involves classes using the *has_classes* and the latter class involves properties using the *has_properties*.

As we shift our view from modelling ontology concepts to the three-level approach method, we find the *Similarity-Searcher* class. This, is an abstraction from the similarity searcher method. It is subclassified into three subclasses *DataTypePropertyAnalysis*, *SpecialProperty-Analysis*, *ClassAnalysis*, each of them representing the analysis method part over the following elements identified in an ontology (Figure 2).

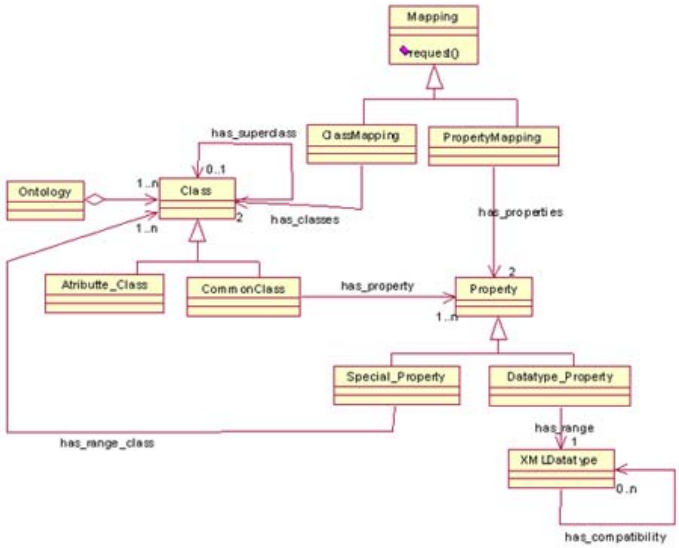


Fig. 5. Plug-in class design diagram part

Similarly, there are in the design plenty of other classes that are needed objects to translate the computer's user inputs to commands to the right objects in the application. For brevity reason we only explained the Similarity Searcher class which is the core of our method.

3.2 Similarity searching

As a result of being too complex to be implemented by a single object, the *SimilaritySearcher* class main responsibility is divided into subresponsibilities reassigned to collaborating objects. Each object implements a quite different similarity search method depending on three elements as above mentioned. The *SimilaritySearcher* object coordinates these collaborating objects, as Figure 6 shows. If the ontology classes to be analyzed are *Common_Class* then datatype and special property analysis would take place, following the class analysis is carried out.

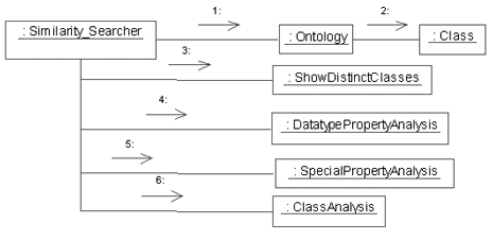


Fig. 6. Symilary_Searcher collaborates with other objects to find similarities.

Those specialized objects, which implement the subordinated responsibilities that the *Similarity Searcher* object delegates as well as coordinates are: *DataType Property Analysis*, *Special Property Analysis*, *Class Analysis*.

Each of these classes implements the different parts of our method. For instance, the DataType Property Analysis component implements the comparison between datatype properties applying all the functions (syntactic and semantic) described in Figure 3.

4 A Case Study - Using the Plugin

Now, we will present a case study that lets us show both the plug-in interface and how the method works. In addition, it let us point out some interesting results depending on the chosed ontologies for mapping. The following two ontologies were selected and created with the Protégé editor, they are shown graphically in Figures 7 and 8.

- The first ontology is called *"Travel Ontolgy"* [1]; it models flights, air agencies, car rental, hotels among other concepts. It has around 40 classes with many properties in order to describe their semantic and structure – they may be special properties or data type properties. From this ontology we only show the *Airport* class and their taxonomical relationships, and properties and restrictions that apply to the related entities, the ones involved in the example.
- The second ontology is called *"Location Ontology"* [2] which possess five classes and a number of properties, both special and datatype properties, to represent a location domain.

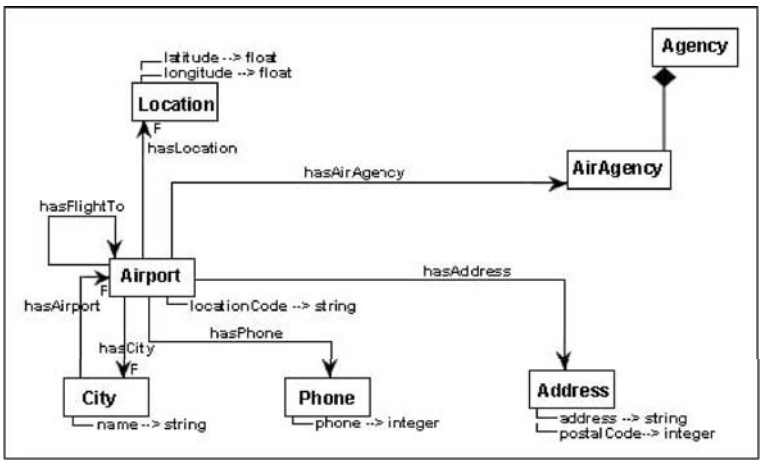


Fig. 7. *"Travel ontology part."*

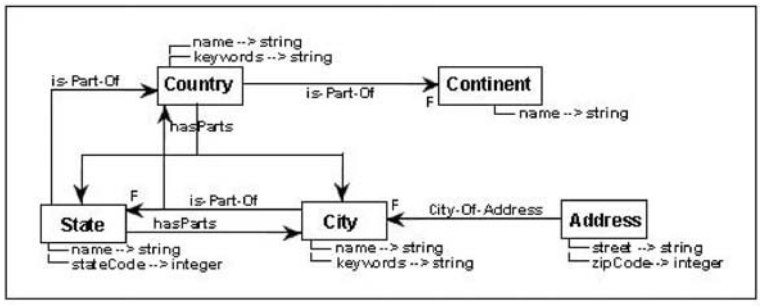


Fig. 8. *"Location ontology part."*

There are some points to notice in the two ontologies. Firstly, in both *Travel* and *Location* ontology graphs there are cycles. It means, that there is a pair of special properties that made

[1] www.ilby.net/travel.owl
[2] http://www.liacs.nl/CS/DLT/pickups/sjoerd/for%20Protege/Science.zip

some class both range and domain class from each other at the same time. In the former, the *hasFlightTo*, *hasAirport* and *hasCity* special properties involving the classes *Airport* and *City* as their ranges and domains. In the other one, for instance, the *hasParts* and *is_Part_Of* involves the classes City and Country as ranges and domains classes from one each other. Nevertheless, our method will detect this and find out a similarity value too. Secondly, regarding to the *City* and *Address* classes modelled in each of the ontologies, they are represented quite different. So the similarity values will not be high in spite of been the same entities. Later we will discuss examples on this points.

Let us now look at the plug-in's interface where we choose the .owl files containing the mentioned ontologies. As Figure 9 shows, this form lets the user to indicate the ontologies to be compared in OWL language.

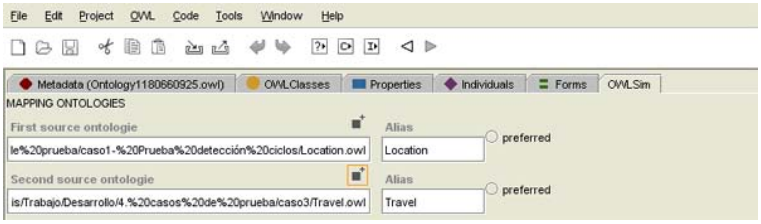


Fig. 9. *"OWLSim plug-in interface which lets us select the owl files to map."*

Following, if both owl files are loaded successfully, the mapping layout screen appears (see Figure 10). It is divided into two main panels. On the left side, there is the *select source classes form* that holds each of the selected ontologies class hierarchy, so that a pair of classes to be compared could be choosen. On the other side, the right one, there is the *show class and property mappings form*. On this window, the mappings between classes and properties found by the procces are shown.

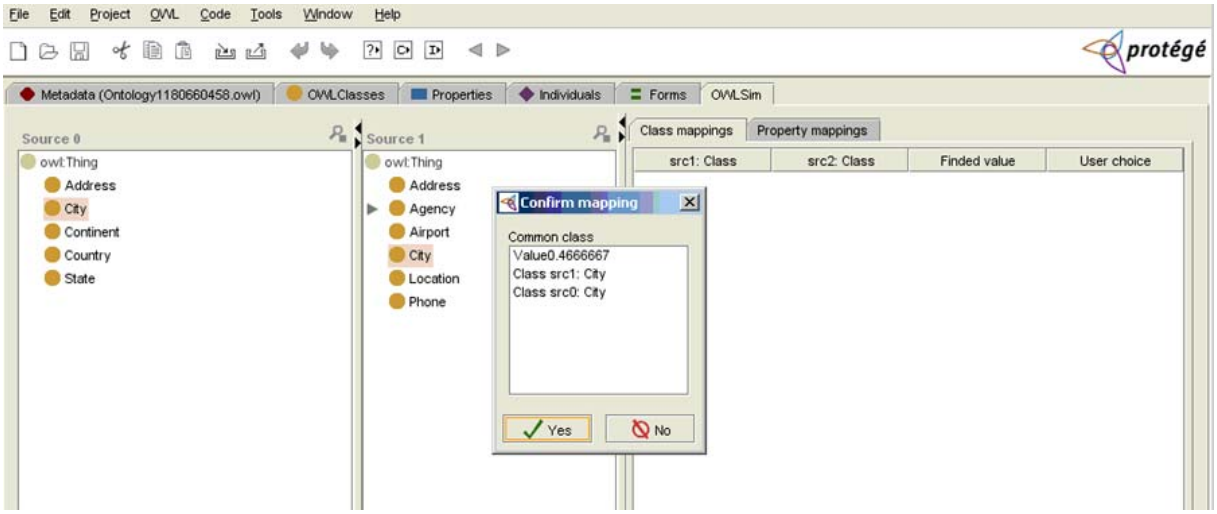


Fig. 10. *"OWLSim plug-in interface shown while the mapping process takes place."*

Once two classes are selected to be compared, the analysis takes place over those selected concepts by clicking the *"Mapp selected"* button. As a result, for any mapping that is found,

a confirmation is required from the user through the confirmation window as shown previously in Figure 10.

Let us now look at two examples, which show the similarity values found when the two points described before occur while using the plug-in. First, when the *Location.City* and *Travel.City* classes are selected to be compared the values shown in Figure 11 will be obtained as the method goes through the special properties graph. As the similarity searching progresses it will find a cycle when the City classes are visited again. The method detects that the node is in the searching path. As a result of the special treatment to this node, we can see that the similarities between the data type properties *cityKeyWords*, *name* y *name_2* (stand for Location.City.name) are calculated. The similarity value from the two last ones only is higher than the acceptance threshold. Moreover, the syntactic and semantic values are equal to one. Thus, the structural comparison regarding the data type properties, *_simAttDtatypeProperties*, is quite low. Moreover, because of being a cycle node, the *_simAttSpecialProperties* is equal to cero. From this, the final value just barely exceeds the acceptance threshold.

```
Go on with the next child: City - City
32. Visited node [City vs City ]
dtp> sim value: cityKeyWords-name==>0.26666668
dtp> sim value: name_2-name==>0.7083334
32. CYCLE IN: City - City
_simThesaurus: 1.0
_simSint: 1.0
_simAttDatatypeProperties : 0.26666668
_simAttSpecialProperties: 0.0
ca-common> sim value: City-City===>0.53333336
```

Fig. 11. *"Similarity values obtained by comparing* City *concepts from both ontologies."*

In the second place, Figure 12 shows the results obtained from the analysis of both *Address* classes. They match syntactically as well as semantically; consequently, the *_simThesaurus* and *_simSint* values are equal to one. However, they are represented quite differently. On the one hand, the *Location.Address* class has two data type properties, *street* and *zipCode* in addition to a special property *city-of-address*. On the other hand, the *Travel.Address* class is represented with two data type properties, *address* and *postalCode*. Although, the address and street properties meant to represent the same information about classes, they are represented rather different. Hence, it will afect the final result. Futher, the comparison of the *zipCode* and *postalCode* properties is the only one that results in a grater value. Consequently, there is just one common attribute between these classes. The final similarity value is higher than the acceptance threshold so a *temporal* mapping is found, and this result will be prompted to the user. Then, if the user confirms this mapping it will be changed to *permanent*.

```
1. Visited node [Address vs Address ]
dtp> sim value: zipCode-postalCode==>0.4625
dtp> sim value: zipCode-address==>0.14583333
dtp> sim value: street-postalCode==>0.14285715
dtp> sim value: street-address==>0.31666666
_simThesaurus: 1.0
_simSint: 1.0
_simAttDatatypeProperties : 0.2
_simAttSpecialProperties: 0.0
ca-common> sim value: Address-Address===>0.47500002
```

Fig. 12. *"Similarity values obtained by comparing* Address *concepts from both ontologies."*

5 Conclusions and Future works

In this paper, we have presented our three level approach for searching mappings between two OWL ontologies, and aspects such as cycles in the ontologies and performance have been considered to avoid possible problems in the application of the method. An implementation of it as a Protégé plug-in has been presented and it will be soon available to be downloaded from Internet [3].

However, currently our work is in a development stage for a number of tasks that are still being developed. Since our current method only deals with one-to-one relationships, we are improving the similarity functions in order to consider many-to-many relationships. Some efforts has been presented in [15].

References

1. Protégé. http://protege.stanford.edu/doc/users_guide/index.html, 2000.
2. L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Addison-Wesley Longman Publishing Co., Inc., Masachusetts, USA, 1998.
3. A. Buccella, A. Cechich, and N. Brisaboa. Ontology-based data integration: Different approaches and common features. In L. Rivero, J. Doorn, and V. Ferraggine, editors, *Encyclopedia of Database Technologies and Applications*. Idea Group, 2005.
4. A. Buccella, A. Cechich, and N. R. Brisaboa. Ontology-based identification of similarity among heterogeneous sources. *Journal of Computer Science and Technology*, 6(1):62–68, 2003.
5. A. Buccella, A. Cechich, and N. R. Brisaboa. A federated layer to integrate heterogeneous knowledge. In *VODCA'04 First International Workshop on Views on Designing Complex Architectures*, number 142 in Electronic Notes in Theoretical Computer Science, Elsevier Science B.V, pages 79–97, Bertinoro, Italy, September 2004.
6. F. Buschmann, R. Meunier, H. Rohnert, P. Sommerland, and M. Stal. *Pattern Oriented Software Arquitecture - A systema of Patterns*. Wiley and Sons Ltd., West Sussex, England, 1996.
7. S. Busse, R. Kutsche, U. Leser, and H. Weber. Federated information systems: Concepts, terminology and architectures. Technical Report Nr. 99-9, Technical University of Berlin, 1999.
8. Martin Fowler and Kendall Scott. *UML distilled (2nd ed.): a brief guide to the standard object modeling language*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.
9. Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.
10. D. Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, July 1998.
11. A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proceedings of the EKAW'02 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 251–263, London, UK, 2002. Springer-Verlag.
12. D. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proceedings of the KR*, pages 483–493, 2000.
13. Natalya Fridman Noy and Mark A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 450–455. AAAI Press / The MIT Press, 2000.
14. M.A. Rodríguez and M.J. Egenhofer. Comparing geospatial entity classes: An asymmetric and context-dependent similarity measure. *International Journal of Geographical Information Science*, 18(3):229–256, 2004.
15. S. Roger, A. Buccella, A. Cechich, and M. S. Palomar. Asematch: A semantic matching method. In *TSD'06: Ninth International Conference on Text, Speech and Dialogue*, pages 229–235, Brno, Czech Republic, September 11-15 2006.
16. M. K. Smith, C. Welty, and D. McGuinness. Owl web ontology language guide. W3C, February 2004.
17. A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.
18. H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hbner. Ontology-based integration of information - a survey of existing approaches. In *Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing*, pages 108–117, Seattle, WA, 2001.
19. R. Wirfs-Brock and A. McKean. *Object Design: Roles, Responsibilities, and Collaborations*. Addison-Wesley 2003, 2003.
20. R. Wirfs-Brock, B. Wilkerson, and L. Wiener. *Designing Object-Oriented Software*. Prentice Hall 1990, 1990.

[3] http://protege.cim3.net/cgi-bin/wiki.pl

A UML Profile for Documenting the Component-and-Connector Views of Software Architectures

Lic. Valerio Adrián Anacleto
Epidata Consulting
Buenos Aires, Argentina, Maipú 521 1ro A
adrian@epidataconsulting.com

Abstract

In this paper, we present a UML profile and a group of UML patterns for documenting the component-and-connector views of software architectures [8]. They facilitate the creation of the component and connector viewtype in any UML 2.0 tool with a compliance level 3 [14]. This work's contributions are: (1) Facilitating the documentation of all the software application's views using only one tool. (2) Curtailing investment in personnel training. (3) Allowing the establishment of an adequate traceability between the architectural artifacts and the rest of the model.

Keywords: software architecture, component-and-connector viewtype, software documentation, UML 2.0

Resumen

En este trabajo presentamos un perfil de UML (UML profile) y un conjunto de patrones de UML (UML Patterns) para documentar la vista de arquitectura de software de componentes y conectores (components and connectors view)[2]. Estos facilitan la creación de la vista en cualquier herramienta UML 2.0 con nivel de conformidad L3 [14]. Este trabajo tiene como aportes: (1) facilitar la documentación de todas las vistas de una aplicación utilizando sólo una herramienta. (1) Reducir la inversión en capacitación de personal. (3) Permitir realizar una adecuada trazabilidad entre los artefactos de la arquitectura y el resto del modelo.

Palabras claves: arquitectura de software, componentes y conectores, viewtype de componentes y conectores, documentación de arquitecturas de software, UML 2.0

1 INTRODUCTION

The aim of this paper is to offer a solution, to a common challenge that arises in practice when an architecture is meant to be documented: how should the component-and-connector views of an application be documented in a syntactic and semantic correct way without losing the traceability of the rest of the documentation artifacts as well as using a unique documentation tool of software applications?

Until its 2.0 version, UML did not count with an appropriate support to document software architectures formally. However, since its 2.0 version, UML has added some new constructs such as: composite structures, ports and roles, which enable the architecture software documentation in a more natural and intuitive way. Although these constructs represent a clear improvement regarding the early UML versions, UML still falls short to document architectures formally [8] and even views as significant as the component-and-connector ones are not easy to document using UML [8].

1.1 Component-and-Connector Viewtype

The component-and-connector viewtype enables the representation of a software architecture from the point of view of its components, the principal unit of runtime interaction or data storage, and its connectors, the interaction mechanism among components and the "data flow" among them [8].

The component-and-connector view is considered one of the most important ones for the developer as well as for the architect [5] and of vital significance for the analysis and quality requirements scope, such as availability, performance, scalability among others.

When trying to design a component-and-connector view we come across a dilemma: should we model the component-and-connector view with an ordinary assistance tool for the UML design tool or should we use an architecture design tool such as: BiZZ design Architect [4], AcmeStudio [1], Aesop [2], Darwin [7] or Unicon [17]? For more information on ADLs and software architecture design tools, refer to [10].

Next, we will analyze some of the consequences of using diverse design tools for different documentation aspects.

1.2 Consequences of Using Different Tools

In everyday practice, the lack of formal knowledge of UML and software architecture as well as the need of books that wipe out the ambiguity in common errors using UML regarding the interpretation of design, analysis and documentation of applications in general already seems to be too much to force the use of multiple tools.

The use of a variety of tools implies a required training to use each tool, which is time consuming and that time means, in turn, an increase in the total cost of ownership (TCO) of the project.

It is feasible that the selected tools have a license cost, which implies a bigger investment in software.

Most companies own a UML design tool but they do not count with one that enables the design of software architectures.

The tools for software architecture design are neither established nor well known in the industry yet. Moreover, the best tools of this type are still only an academic initiative and, regretfully, the academic-industrial gap is big, and at the same time, the usability and quality of the tools are not the best.

Traceability of the documentation elements: for us, this is the most important of all the problems because of its impact on the usability and money expense as regards the documentation maintenance cost. By traceability we understand an existent relationship between the model elements. There can be different types of traceability relationships according to the particular requirements. For instance, the trace of the links among a diagram's elements or the trace that shows the evolution from the requirements to the final code, linking all the artifacts in between which, in a development process, usually represent the abstraction level and maturity of the solution in a specific stage of the development process. For instance, we could trace a group of requirements that gave life to a use case; at the same time, this use case could be traced with an analysis diagram and so on until we reach the code that implements the functionality of this use case.
Having an appropriate traceability in a model allows, among other things, to analyze dependencies, to estimate the impact in the changes of an artifact, and to distribute the work and analyze the system's quality attributes by means of, for example, traces between the software components and the nodes where the deployment will take place.

By having two separate modeling tools for the architecture and for the rest of the application, the key traceability elements that create the links with the architecture are lost; therefore, they have to be kept separately, for instance, by means of using a traceability matrix.

1.3 Advantages of Using Different Tools

It is useful to use a variety of tools if, due to the characteristics of the application or the maturity degree of the software architecture practices, it is necessary to document that architecture using the highest possible level of detail and strictness.

Throughout a variety of software architecture assessments, we have discovered that, regardless of the tool and the type of view to document, it is imperative to count with a software architecture documentation that shows, among other things: the most important processes, the components that make up the architecture (sometimes called in industry "Architecture Map"), their dependencies and coupling, and in which abstraction level they occur (data, business logic, etc.).

It is striking that, in a quite high percentage, there are not many companies counting with a software architecture that meets these minimal conditions.

We believe that, in order to achieve this goal, it suffices to use any UML modeling tool in an adequate way, documenting the component-and-connector views within the same tool by means of a UML profile specification [14].

In this paper we present a UML profile by means of which the component-and-connector views in any UML modeling tool that supports a compliance level - complete (L3) can be documented, avoiding the consequences of using a variety of modeling tools.

The rest of the work is organized in the following way: in the first part we present our choice for modeling the component-and-connector views using UML 2.0 and representing it by means of UML patterns and profiles. In the second section we show a design, as an illustrative example, using the developed UML profile. Then, we analyze the work done from the point of view of

usability. Finally, we make comments about some possibilities of future work and provide conclusions on the current paper.

2 MODELING THE COMPONENT-AND-CONNECTOR VIEWTYPE WITH UML 2.0

In order to facilitate the design of the component-and-connector views we have developed a UML profile [14] which, by being a standard specification defined by the OMG (Object Management Group), guarantees us that it will be able to be used by any tool that implements UML 2.0 or higher. The need to use the 2.0 or a higher UML version is due to the fact that in the 2.0 version some documentation constructs are introduced, such as ports and roles, which are useful for our UML profile.

Garlan's work [8] shows several of the available options for modeling a component-and-connector views using UML 2.0. We took as a basis the analysis done in that work for making decisions when creating our own UML profile. It is also important to mention that some of the decisions are original from this work; they will be described in detail later on.

Next, we will list the elements that make up the UML profile for the component-and-connector viewtype.

2.1 Components

We decided to document the components using the "Component" documentation construct defined in UML 2.0. Another option to document them was using the "Classes" documentation construct. There are some opinions that claim that the latter could vanish from the future UML versions since the semantics of both artifacts overlap considerably [12].

We decided to give the component a similar visual aspect to the one used in Christine Hofmeister's book [11] when documenting architecture diagrams, because we found it intuitive and practical for the purpose of documenting a software architecture.

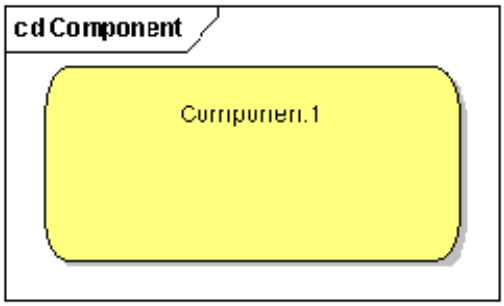


Figure 1: visual aspect of a component

2.2 Connectors

In all the papers and significant documents on software architecture documentation, the importance of treating connectors as "first class citizens" [15] is highlighted; in UML it could be expressed as a classifier and not as a simple association. The fact that a connector appears as a classifier has very deep implications in the expressive power of the connector and in the traceability of the artifacts; for example, one could create a connector as a structured classifier and within that connector define

the class diagram related to the design itself, its sequences, collaborations and quality requirements specifications, which would greatly facilitate the analysis of quality attributes that may or may not be reached by a software architecture.

In spite of the fact that we take Garlan's work [8] as a fundamental reference for our current work; the former does not include as an option the use of a "component" construct to document a connector. We believe that a connector's semantics is much more similar to the one of a component than to an association or an association class, which are the options mentioned in the quoted work. We believe this because UML component are "first class citizen" [15] meanwhile association and association class aren't independent classifiers.

That is why in our UML profile we use as a basis a UML component to document a connector (from the component-and-connector viewtype). By changing its appearance by means of a stereotype, we can distinguish it visually from a component. We follow the visual aspect defined by Christine Hofmeister [11].

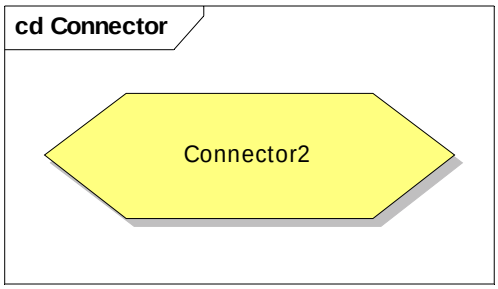


Figure 2: visual aspect of a connector

2.3 Ports

Ports are constructs defined in UML 2.0; therefore, we use them just as they are defined in the standard itself. Taking into account the fact that a port can only belong to a component, this restriction is not validated in the UML profile since not all the tools support OCL adequately.

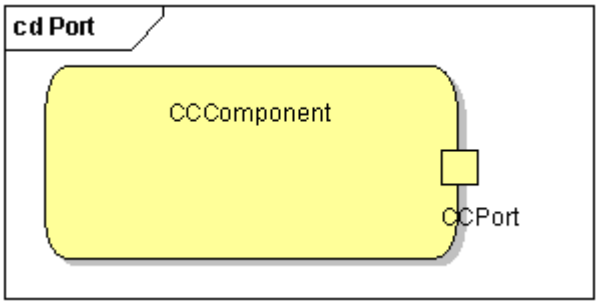


Figure 3: visual aspect of a port associated to a component

2.4 Roles

Just like in the case of ports, roles are defined in UML 2.0; therefore, we believe their use is quite convenient. However, a role will only be associated to a connector kind of construct.

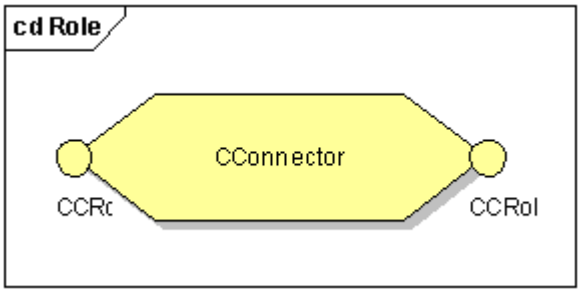


Figure 4: visual aspect of two roles associated to a connector

2.5 Association

To associate ports and roles, we use an association defined in our UML profile for the purpose of distinguishing it from other types of associations and allowing the subsequent development of tools that gain benefits and maximize the use of the current UML profile.

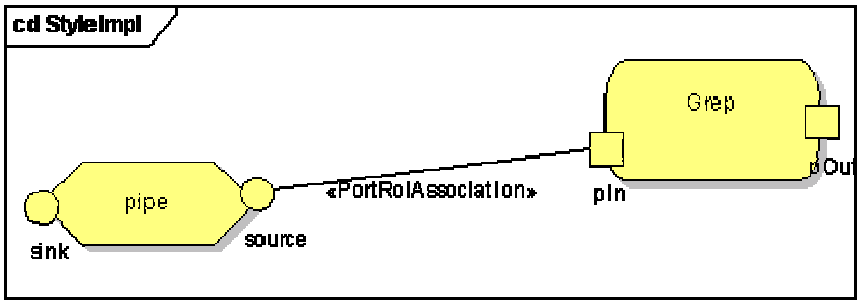


Figure 5: visual aspect and stereotype of the port-and-role association

2.6 Delegation

We thought it would be convenient to create a delegate association to distinguish it from the ordinary association. A delegate association only takes place between roles and it corresponds to the delegation of a message received by a port, which can be called “A” and which delegates the message to another one: port “B”. In this case it could be said that port “A” delegates the message to port “B”. We identify this association by placing an arrow on one of the ends of the line to indicate the message direction.

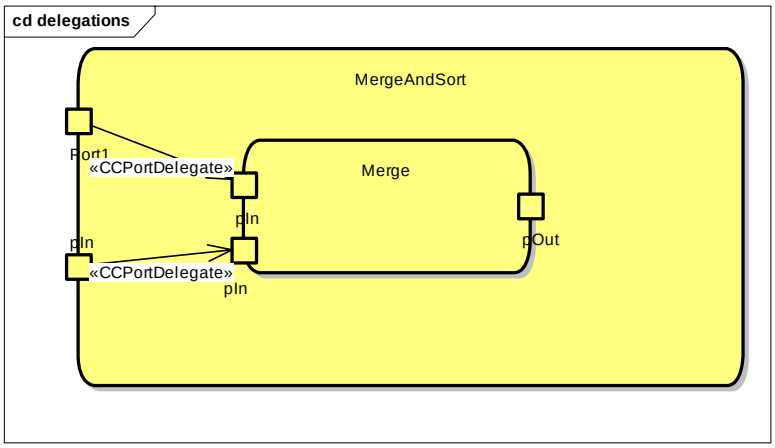


Figure 6: visual aspect and stereotype of two delegations

2.7 Properties

Our decision for documenting software architecture properties is to use tagged values, validated for the components' instances. In the case of properties shared by all the instances (type properties), we use attributes.

3 THE UML PROFILE

Next, we show the design diagram of the UML profile we have developed. It can be downloaded from [6]. It is important to mention that it was only used with the Enterprise Architect tool, which can be downloaded from [16]. Since a UML profile is a standard defined by the OMG, any tool that keeps to the standard should be capable of using the UML profile. Other tools that abide by the standard according to the OMG can be found in [18].

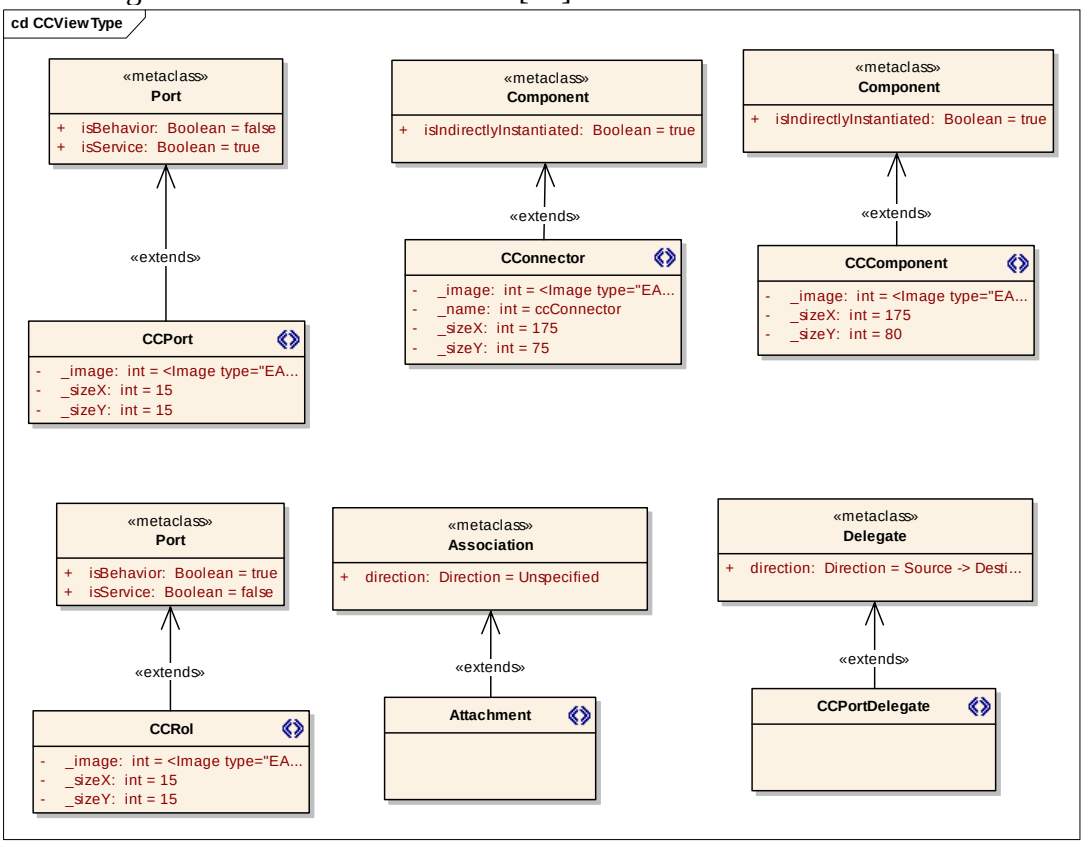


Figure 7: profile design

4 AN EXAMPLE ON HOW TO USE THE PROFILE

In figure 8 we show an example in which we document a pipe-and-filter sequence; the sequence itself is of no significance, but it is important to highlight the power of the developed profile. It represents instances of some components and connectors defined in figure 9.

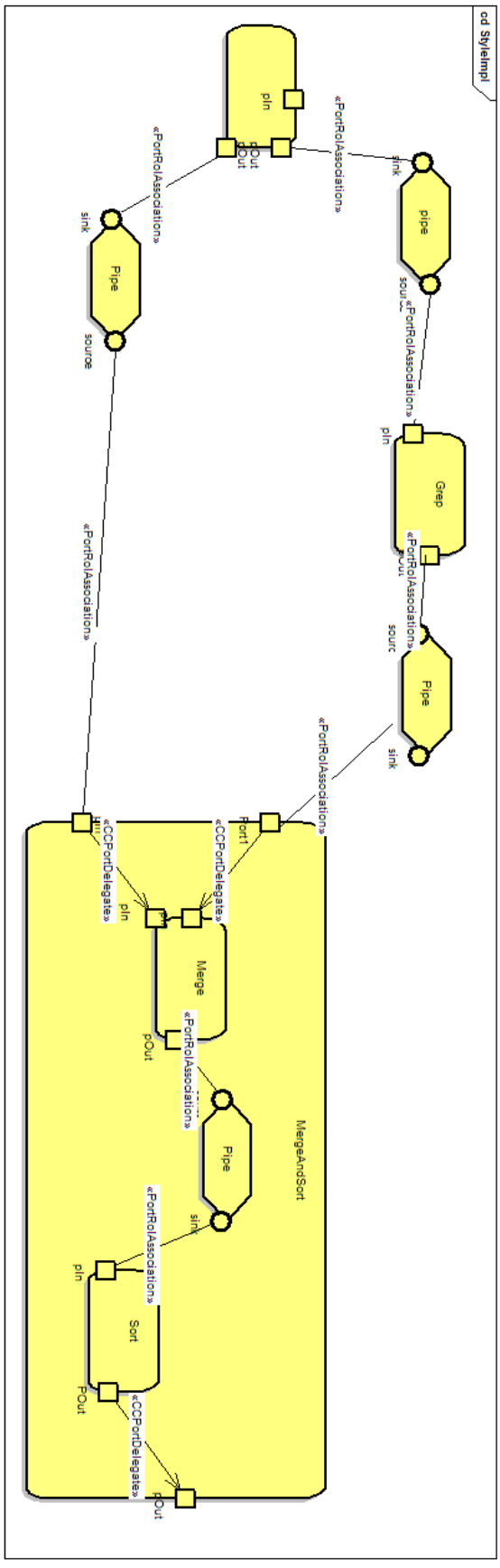


Figure 8: sequence of a pipe-and-filter view

The former diagram was created using instances of an architectural type diagram, which can be found in figure 9 and which represents the design of the components and connectors for a pipe-and-filter architecture style as well as their possible relationships, for instance, it can be noticed that a filter is related to a pipe through the ports of the first and the roles of the second and that “grep”, “merge”, “sort” and “splitter” are specializations of “filter”, and as a consequence, they inherit its semantics.

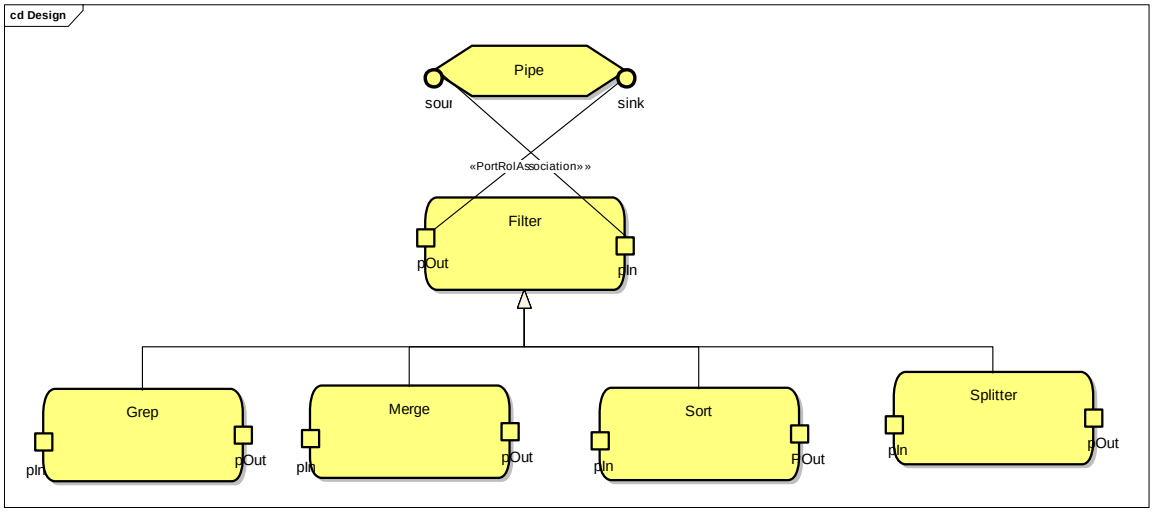


Figure 9: definition of architectural types for a pipe-and-filter architectural style

5 USABILITY

The International Standardization Organization (ISO) offers two definitions of usability:

ISO/IEC 9126: “Usability refers to the software’s ability to be understood, learned, used and considered attractive by the user, under specific conditions of use.”

This definition emphasizes the internal and external attributes of the product —regardless whether it is a software or not— which contribute to its functionality and efficiency. The usability depends not only on the product itself but also on the user. That is why a product is by no means usable by itself; it can only be used within a specific context and by specific users. The usability cannot be valued if a product is studied in an isolated way.

ISO/IEC 9241: "Usability is the efficiency and satisfaction with which a product enables specified users to achieve specified goals in a specified context of use”.

This definition focuses on the concept of quality of use, that is to say, it refers to the way the user performs specified tasks under specified circumstances effectively.

5.1 Usability of the UML Profile

It seems to be clear that, for the profile to be useful, it is necessary that it is usable in the sense given by both definitions above.

We implemented the profile bearing in mind the graphic aspects and following a well known component-and-connector metaphor taken from Hofmeister’s book [11], which bears resemblance to the one used in many works by Garlan, Shaw and other software architecture precursors and also

to the iconography and metaphor used in some software architecture documentation tools, such as [17] and [1]. This makes the choice we have presented here intuitive.

There is a second aspect which has a deep impact in usability: it is that every component will have, at least, one port and every connector will have, at least, one role. The user will find it repetitive and tedious to drag and drop a port or a role every time he documents a component or a connector. To improve this aspect, we decided to use another standard defined within UML and implementing some component-and-connector design patterns. These patterns can be, for instance and among others, a component with a port, as shown in figure 3, a component with two ports, a connector with two roles, and so on.

This would considerably facilitate the use of the tool, making it more agile by avoiding repetitive tasks without losing the formality and semantics of components and connectors.

5.2 Integration of the UML Profile with Commercial Tools

In figure 10 we show the use of the UML profile with the Enterprise Architect tool. In the toolbox on the left, the menu is restricted for the component-and-connector viewtype with the elements defined in the UML profile. In the resource view on the right, the patterns that facilitate the use of the defined profile can be appreciated.

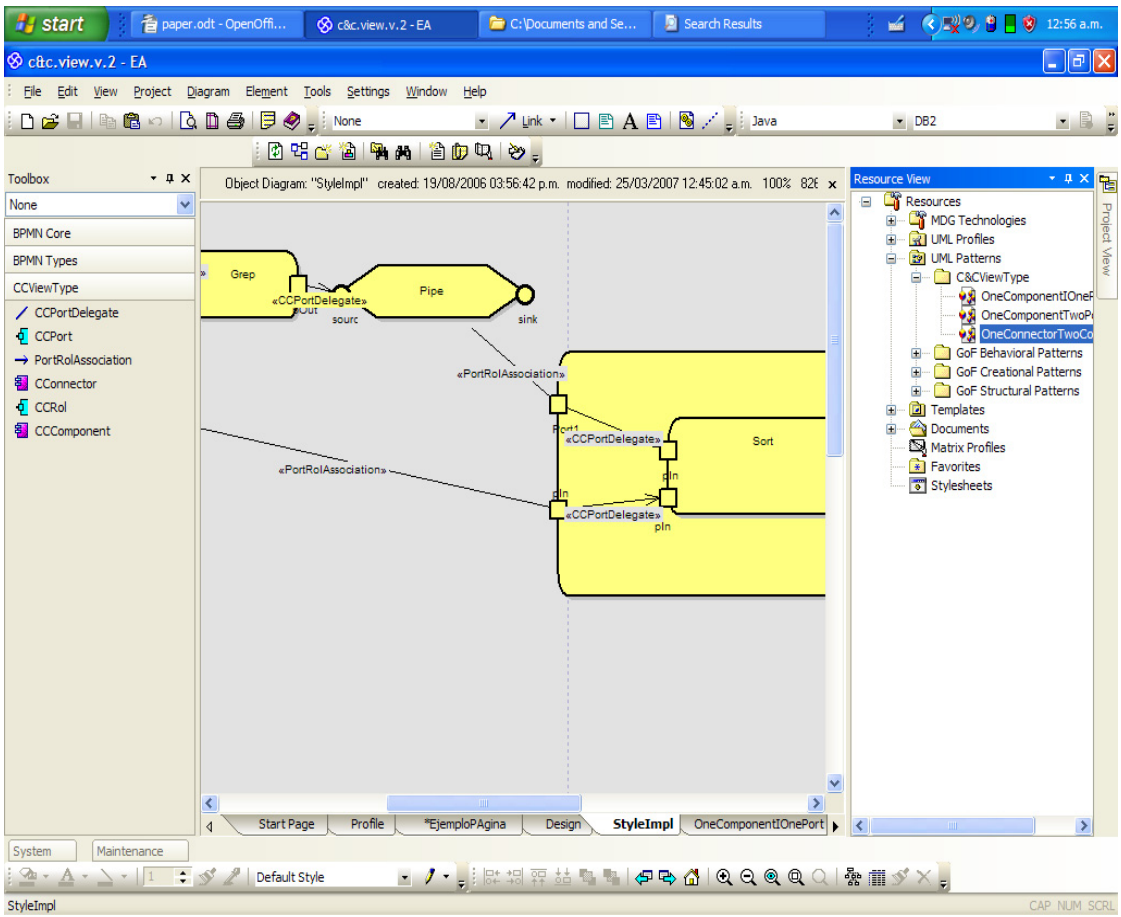


Figure 10: integration of the UML profile with commercial tools

6 FUTURE WORK

In future works, it would be interesting to look for the way to create traces between composite elements as if they were only one element, for example, a component and its ports. This would enable the traceability of a component with its ports and inner structure as a single element instead of many.

Integrating this profile with an ADL would be another possibility of future work, though it depends on OMG choosing an ADL for the standard UML, in case any decision is taken on the short run.

7 CONCLUSIONS AND CONTRIBUTIONS

In the current work, we implemented a UML profile for documenting the component-and-connector views. For the implementation of the profile, our main reference was Garlan's work [8], though it was also necessary to take some decisions that were not taken into account in that work.

We believe that using this profile to document the component-and-connector views will facilitate the software architects' work by enabling the documentation of the architecture together with the rest of the application's design. By facilitating and, as a consequence, disseminating the use and practices of architecture, since a UML profile is a standard, it can be used with any UML design tool, diminishing the complexity of having to deal with a variety of tools and the overall cost of documenting a software application in an appropriate way.

REFERENCES

[1] Acme Project – home page: http://acme.able.cs.cmu.edu/index.html

[2] Aesop - home page: http://www.cs.cmu.edu/~able/aesop/aesop_home.html

[3] Len Bass, Paul Clements, Rick Kazman. Software Architecture In Practice, Second Edition. Boston: Addison-Wesley, p. 21-24. ISBN 0-321-15495-9, 2003.

[4] BiZZdesign Architect – home page: http://www.bizzdesign.nl/html/bizzdesignarchitect_en.html

[5] Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford (2003). Documenting Software Architectures: Views and Beyond. Boston: Addison-Wesley, pp. 13-15. ISBN 0-201-70372-6.

[6] Component and Connector profile, patterns and sample downloads: http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=A+UML+Profile+for+Documenting+the+Component+and+Connector+Views+of+Software+Architectures+Downloads

[7] Darwin, The Software Architect's Assistant – home page: http://www.doc.ic.ac.uk/~kn/java/saaj.html

[8] David Garlan, James Ivers, Paul Clements, Robert Nord, Bradley Schmerl and Jaime Rodrigo Oviedo Silva. Documenting Component and Connector Views with UML 2.0. Technical report, CMU/SEI-2004-TR-008, Software Engineering Institute, 2004.

[9] David Garlan, R. Monroe, and D. Wile. "Acme: an Architecture Description Interchange Language," Proceedings of the 1997 Conference of the Centre for Advanced Studies on Collaborative Research, ACM, New York (1997), p. 7.

[10] David Garlan. Software Architecture: a Roadmap, in The Future of Software Engineering, A. Finkelstein, Ed.: ACM Press, pp. 93-101, 2000.

[11] Christine Hofmeister , Robert Nord , Dilip Soni. Applied software architecture, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1999

[12] C. Kobryn. UML 3.0 and the Future of Modeling. Software System Modeling, 3:4--8, 2004.UML3.0

[13] MetaObject Facility (MOF) 2.0 Core Specification, Available Specification, OMG document ptc/04-10-15, Object Management Group (2004), http://www.omg.org/ cgi-bin/doc?ptc/2004-10-15.

[14] UML 2.0 Superstructure Specification, OMG document formal/05-07-04, Object Management Group, Inc. (2005), http://www.omg.org/cgi-bin/doc?formal/05-07-04

[15] Mary Shaw and David Garlan. Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall, 1995

[16] Sparx Systems - Enterprise Architect home page: http://www.sparxsystems.com.au/

[17] Unicon – home page: http://www.cs.cmu.edu/People/UniCon/

[18] UML Directory: http://uml-directory.omg.org/

Algoritmo de Reunión Espacio-Temporal usando estructura 3DR-tree podada

Anibal J. Díaz [*]
Departamento de Informática, UTN Universidad Tecnológica Nacional
Concepción del Uruguay, Entre Ríos - Argentina
anibaljdiaz@yahoo.es

and

Gilberto A. Gutiérrez [**]
Departamento de Auditoría e Informática
Universidad del Bio-Bio - Chillán, Chile
ggutierr@ubiobio.cl

and

Olinda E. Gagliardi
Departamento de Informática - Facultad de Ciencias Físico-Matemáticas
Universidad Nacional de San Luis - Argentina
oli@unsl.edu.ar

Abstract

The join query is one of the most important operation in the Database Management System (DBMS). Because of the modern (DBMS) have incorporated new types of data bases (temporal, spatial, among others), it has become necessary to possess new algorithms to process join queries in these areas. Nowadays the databases have been receiving major importance due to the wide variety of applications that they need to process the spatial history of the objects. The spatio-temporal access methods proposed till now are orientated to trying principally timeslice and interval query and they do not approach the join query. In this work we present and evaluate a join space-temporal algorithm based on it structures 3DR-tree and that benefits by the applied pruned technique from. Also it is introduced an analytical model about the algorithm that estimate the cost in terms of node or disk accesses. Finally the predictive behavior of the proposed model will be compared with the 3DR-tree cost model showing the accuracy of this model.

Keywords: Spatiotemporal Join (STJ), 3DR-tree, Reunión espacio temporal, Modelo de costo, 3DR-tree podado.

[*]Este trabajo es parcialmente subvencionado por los Proyecto de Tecnologías Avanzadas de Bases de Datos 22/F314, Departamento de Informática, UNSL; y el Proyecto AL06-PF-013 Geometría Computacional.

[**]Parcialmente financiado por el proyecto Algoritmos para la evaluación de consultas espacio-temporales, código 073218 A/R de la Universidad del Bio-Bio.

Resumen

La consulta de reunión (join) es una de las operaciones fundamentales en un Sistema de Administración de Bases de Datos (DBMS). Debido a que los (DBMS) modernos han incorporado nuevos tipos de bases de datos (temporal, espacial, entre otros), se ha hecho necesario contar con nuevos algoritmos para procesar consultas de reunión en esas áreas. Actualmente las bases de datos espacio-temporales han ido cobrando mayor importancia debido a la amplia variedad de aplicaciones que requieren procesar la historia espacial de los objetos. Los métodos de acceso espacio-temporales propuestos hasta ahora estan orientados a procesar principalmente consultas de tipo time-slice y de tipo interval y no abordan la consulta de reunión. En este trabajo presentamos y evaluamos un algoritmo de reunión espacio temporal basado en la estructuras 3DR-tree y que se beneficia por la técnica de podado aplicada. También se introduce un modelo analítico del algoritmo que estima el costo en términos de accesos a nodo o accesos a disco. Finalmente se evalua el comportamiento predictivo del modelo propuesto con el modelo original de costo del algoritmo de reunión espacio-temporal 3DR-tree mostrando la precisión del mismo.

1 Introducción.

1.1 Conceptos preliminares.

Los objetos espacio-temporales, esto es objetos cuyo atributo espacial cambia a través del tiempo, aparecen en muchas aplicaciones [5,9,11]. La naturaleza de tales aplicaciones requiere del procesamiento de grandes cantidades de datos espacio-temporales generados de manera muy frecuente. El procesamiento eficiente de consultas, sobre estos volúmenes de datos, demanda de nuevos, sofisticados y eficientes esquemas de indexación o métodos de acceso espacio-temporales. Los métodos propuestos hasta ahora apuntan principalmente a resolver dos tipos de consultas espacio temporales conocidas como time-slice y time-interval [12]. El primer tipo recupera todos los objetos que intersectan un rango espacial, en un instante de tiempo especifico, en cambio el segundo extiende la idea a un intervalo de tiempo. Otro tipo de consultas en ambientes espacio-temporales de reconocida importancia en [8,11,10] es el de reunión para el cual, de acuerdo a lo que conocemos, no existen esquemas de índices ni algoritmos, salvo 3DR-tree [20] en donde la reunión espacio-temporal corresponde a una reunión espacial considerando el tiempo como otra dimesión espacial como condición indispensable para su procesamiento eficiente.

1.2 Definición del Problema.

La operación de reunión espacial, $S1 \bowtie_\theta S2$, combina dos conjuntos de objetos espaciales S1 y S2, basado en el predicado θ. En el ámbito espacio temporal se puede definir la consulta de reunión espacio-temporal como $S1 \bowtie_{\theta \wedge \tau} S2$, pero ahora S1 y S2 son dos conjuntos espacio-temporales, donde τ corresponde a un predicado temporal y θ al espacial.
Por lo tanto y de acuerdo a [8] *"Dadas dos relaciones espacio-temporales $S1$ y $S2$, la consulta de reunión espacio-temporal se define como la operación de hallar los pares de objetos que pertenecen a ambos conjuntos y cuyas extensiones se intersectaron durante el intervalo de tiempo t "*

De esta forma podemos responder consultas como : *Encontrar todos los autos que tuvieron una colisión el dia de ayer*. Suponiendo que el conjunto contiene todas las posiciones de los autos, la consulta anterior implica buscar todos los autos cuyas trayectorias se intersectaron en el intervalo de tiempo consultado. Claramente esta consulta se resuelve por medio de una reunión espacio temporal.

2 Antecedentes.

En el modelo de dato espacio-temporal, los objetos espacio-temporales son aquellos que pueden cambiar su forma y su posición en el espacio, o cambiar su forma sin cambiar la posición o cambiar su posición y no su forma. En este campo, y tomando como ejemplo objetos en movimiento cuya geometria propia no se modifica, la operación de reunión espacio-temporal tiene como antecedente la propuesta por [1] y que se aplica sin modificaciones en la estructura de dato 3DR-tree. Como nuestro algoritmo de reunión espacio-temporal se basa en este método de acceso, en esta sección describiremos el mismo junto con el algoritmo de reunión para dicha estructura y su modelo de costo.

2.1 Método de acceso espacio-temporal 3DR-tree

3DR-tree, es un método de acceso espacio-temporal que considera el tiempo como una dimensión adicional a las coordenadas espaciales. De esta forma cuando un objeto, en un espacio de 2 dimensiones, cambia su posición o forma geométrica se inserta un MBR formado por 2 puntos en 3 dimensiones. Un MBR es un "rectángulo mínimo envolvente", el cual es capaz de contener los atributos espaciales del objeto. Este rectángulo mínimo es la aproximación geométrica de un objeto real, que se guarda en el nodo hoja del árbol, y los nodos internos representan una sucesión de regiones rectangulares minimales, cada una de las cuales controla un nodo en el nivel inferior.

La estructura 3DR-tree es muy eficiente en el uso del almacenamiento y en el procesamiento de consultas de tipo time-interval. Sin embargo, una de sus principales desventajas es su ineficiencia para consultas de tipo time-slice debido a la gran altura que puede alcanzar provocada por la gran cantidad de MBR's que se deben insertar. Esta deficiencia se reduce considerablemente gracias a la técnica de podado que se aplica a la estructura de indexación como se explica en la Sección 3.

2.2 Algoritmo de reunión espacio-temporal basado en 3DR-tree

Dado que un 3D R-tree es un R-tree en un espacio de 3 dimensiones, la implementación de la consulta de reunión espacio-temporal se resuelve en forma directa utilizando el procedimiento propuesto en [1] para la reunión espacial y que se muestra en el *Algoritmo 1*. Dicho algoritmo supone que los dos conjuntos de objetos tienen disponible un índice espacial basado en R-tree. Cabe destacar que el procesamiento de la reunión espacial considera dos etapas: filtrado y refinamiento. En la etapa de filtrado se obtiene un superconjunto de objetos los cuales son seleccionados de acuerdo a su aproximación espacial(MBR). Posteriormente, en la etapa de refinamiento (utilizando la geometría exacta de los objetos) se verifica si se cumple o no el predicado espacial. El procedimiento descrito por el Algoritmo 1 (**SJ**) se utiliza para la etapa de filtrado y funciona tanto si los árboles tienen igual o diferente altura.

2.3 Modelo analítico para la reunión espacial usando R-tree

Ya que 3DR-tree es utilizado para comparar nuestro algoritmo de consulta de reunión espacio-temporal, es que en esta sección describimos el modelo de costo que se plantea para la reunión espacial. Primero explicamos un modelo para consultas espaciales de tipo "window query" y luego para la consulta de reunión espacial.

Existen varias propuestas para estimar el rendimiento de una consulta espacial (windows query) sobre un R-tree [6,10,13,16,18,19]. Uno de los modelos más conocido es el descrito en [16,19].

En [19] se propone un método para estimar el costo de una operación de reunión espacial. Dicho modelo asume un espacio multidimensional y que existen dos conjuntos espaciales de cardinalidad N_{R1} y N_{R2} respectivamente y que cada uno de ellos cuenta con un R-tree, $R1$ y $R2$. El objetivo del modelo es obtener el número promedio de nodos (NA) accesados o número promedio de bloques de disco leidos (DA) por una operación de reunión espacial. La diferencia entre NA y DA se deben a que en el cálculo de NA no se

Algorithm 1 Algoritmo para procesar Reunión espacial de objetos espacio- temporales

```
1: SJ(R1, R2) {R1 y R2 corresponden a nodos de R-tree(3DR-tree)}
2: for all E1 in R1 do
3:    for all E2 in R2 do
4:       if IntersectMBR(E1.MBR, E2.MBR) then
5:          if R1 y R2 son hojas then
6:             output(E1.Oid, E2.Oid)
7:          else if R1 es hoja then
8:             ReadPage(E2.ptr)
9:             SJ(E1.ptr, E2.ptr)
10:         else if R2 es hoja then
11:            ReadPage(E1.ptr)
12:            SJ(E1.ptr, E2.ptr)
13:         else
14:            ReadPage(E1.ptr)
15:            ReadPage(E2.ptr)
16:            SJ(E1.ptr, E2.ptr)
17:         end if
18:      end if
19:   end for
20: end for
```

considera un sistema de buffer mientras que para DA si. De esta forma la desigualdad DA $\leq$ NA siempre se cumple. En primer lugar, suponiendo que R1 y R2 tienen la misma altura h y que las dos raíces se almacenadas en memoria, el valor de NA según[19] se define en la Ec.(1) :

$$NA_{total}(R_1, R_2) = \sum_{j=1}^{h-1} \{NA(R_1, j) + NA(R_2, j)\} \tag{1}$$

donde $NA(R_i; j)$ es el número promedio de nodos en el nivel j del R-tree i y está dado por :

$$NA(R_i, j) = N_{R_2,j}.N_{R_1,j} \prod_{k=1}^{n} min\{1, (s_{R_1,j,k} + s_{R_2,j,k})\} \tag{2}$$

y según [18] el costo total de la reunión espacial vale:

$$NA(R_1, R_2) = \sum_{j=1}^{h-1} N_{R_1,j} \left(N_{R_2,j} \prod_{k=1}^{n} [S_{R_1,j,k} + S_{R_2,j,k}] + 1 + N_{R_2,j+1} \left[\prod_{k=1}^{n} (S_{R_1,j,k} + S_{R_2,j+1,k}) \right]^2 \right) \tag{3}$$

Pero en la expresión para la cantidad de nodos promedio por nivel debemos tener en cuenta el rango de crecimiento de los mismos a través del tiempo en la estructura. Luego debemos considerar el efecto del tiempo como lo expresa [14]:

$$\mathbf{N_{i,T}} = \mathbf{N_{i,0}} + \mathbf{N_{i,0}} \cdot (\mathbf{T} - \mathbf{1}) \cdot \mathbf{a}_* \tag{4}$$

donde $\mathbf{N_{i,T}}$ es el número total de objetos del árbol **i** al cabo de un tiempo **T**, $N_{i,0}$ es el número incial de objetos en el árbol i (T=0) y a_* es el rango promedio de crecimiento en la estructura. En caso de haber variacion de la forma del objeto con el paso del tiempo en $\mathbf{a}_*$ se debe considerar el promedio de cambio de forma a_{ms} con lo cual $\mathbf{a}_* = \mathbf{a_{ext}} + \mathbf{a_{ms}} - \mathbf{a_{ext}} \cdot \mathbf{a_{ms}}$ [14].

3. Algoritmo de Reunión espacio-temporal usando 3DR-tree podado.

3.1 Estructura espacio-temporal APR-tree.

La estrategia de particionar el árbol principal de indexación la aplicamos para reducir su altura, generando varios árboles 3DR-tree-podados. Un árbol 3DR-tree-podado es una estructura arborea 3DR-tree pero de

menor altura la cual se obtiene dividiendo la estructura principal en múltiples 3DR-trees responsables cada uno de un intervalo temporal $L_i < T$.

Una idea similar es la que se propone en [7] donde dividen el árbol principal en árboles más pequeños de acuerdo con el tamaño de la carga de trabajo, llamándo al resultado de la partición estructura APR-tree. En esta estructura el intervalo temporal $\mathbf{L_i}$ que administra cada 3DR-tree-podado depende de la carga de trabajo de la estructura principal. En la Figura 1 se puede observar las longitudes temporales L_1, L_2 y L_i de las estructuras R_1, R_2 y R_i del APR-tree.

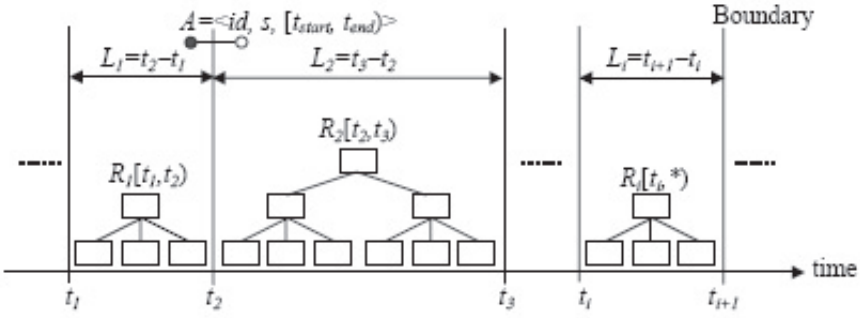


Figura 1: Estructura APR-tree

El APR-tree no comparte nodos, pero el particionado del árbol principal genera duplicados en los árboles podados. Una entrada tiene la forma $A\langle id, s, [t_s, t_e)\rangle$, donde id es el identificador del objeto, s es el MBR espacial, y t_s, t_e son los instantes en que el objeto se ha ingresado a la base de datos y en que se ha modificado. Cuando una entrada por ejemplo A_1 intersecta el límite del intervalo temporal L_1 entre las estructuras R_1 y R_2, se divide el lifespan de la entrada en dos y obtenemos $A_{11}\langle id, s, [t_s, t_2)\rangle$ y $A_{12}\langle id, s, [t_2, t_e)\rangle$ insertándo A_{11} en R_1 y A_{12} en R_2. Esto es una limitación a la hora de realizar la búsqueda en los nodos para determinar los pares de objetos que se solapan. Sin embargo esta limitación se ve minimizada gracias a la reducción en altura obtenida en la división de la estructura principal y por la gran reducción de espacio muerto obtenida por el fraccionamiento de los datos.

3.2 Nuestra propuesta .

Sean los conjuntos S1 con N1 objetos espacio-temporales con una historia T1 y S2 con N2 objetos y una historia T2. Aplicando la metodología de partición de [7] obtenemos K1 y K2 árboles podados de cada conjunto y cuyas raíces e intervalos temporales guardamos en arreglos A y B respectivamente, es decir:
$A = \{S1_1, S1_2, S1_3, ..., S1_{K1}\}$ y $B = \{S2_1, S2_2, S2_3, ..., S2_{K2}\}$
Siendo los intervalos temporales de cada conjunto :
$T1 = \{L1_1, L1_2, ..., L1_{K1}\}$ y $T2 = \{L2_1, L2_2, ..., L2_{K2}\}$
Una vez finalizado el paso anterior se define el intervalo de vida promedio por conjunto, $L1$ y $L2$ como $L1 = \frac{\sum_{i=1}^{K1} L1_i}{K1}$, donde $(L1_i \cap L1_{i+1}) = \emptyset$ y $\bigcup_{i=1}^{K1} L1_i = T1$, análogamente para L2.

Partiendo de los arreglos A y B pasamos a procesar la reunión implementando el *Algoritmo 2* sobre los pares de estructuras de A y B. Este algoritmo utiliza la información de que los conjuntos A y B se encuentran ordenados por el tiempo. El algoritmo comienza eligiendo el intervalo más antiguo entre los conjuntos A y B. Luego verifica todos aquellos intervalos que se intersectan con él y cumplan que el límite inferior del intervalo que intersecta sea inferior al límite superior del intervalo más antiguo. Se verifica entonces que los 3DR-tree-podados responsables de esos intervalos, se intersecten espacialmente y si cumplen esa restricción entonces se procesa la reunión espacial simple (*Algoritmo 1*: SJ(RA_i,RB_j)). En este algoritmo se modifica el final de la línea 4, agregando $AND(E_2.t_e \leq R1.t_e)$ $AND(E_1.lifespan \cap E_2.lifespan)$, a efectos de tener en cuenta la existencia de duplicados generados por la metodología de partición de la estructura principal

y para verificar que las entradas cumplen con el predicado de la reunión espacio temporal. A continuación se selecciona el siguiente intervalo más antiguo y se repite el proceso hasta alcanzar el límite $K1$ o $K2$ del menor de los conjuntos tratados.

Algorithm 2 Algoritmo de reunión mejorado sobre arboles podados de A y B

```
1: Filtrar(A,B) {RA_i y RB_j son 3DR-tree-podados de A y B respectivamente}
2: i=0; j=0;
3: while (i < K1 and j < K2) do
4:    if (RA_i.t_s ≤ RB_j.t_s) then
5:       if (RA_i.t_e > RB_j.t_s) then
6:          if IntersectMBR(RA_i.MBR, RB_j.MBR) then
7:             SJ(< RA_i, RB_j >)
8:          end if
9:          Antiguo(RA_i, RB_j)
10:      else
11:         i++;
12:      end if
13:   else
14:      if (RB_j.t_e > RA_i.t_s) then
15:         if IntersectMBR(RA_i.MBR, RB_j.MBR) then
16:            SJ(< RA_i, RB_j >)
17:         end if
18:         Antiguo(RA_i, RB_j)
19:      else
20:         j++;
21:      end if
22:   end if
23: end while
```

Algorithm 3 Algoritmo que mantiene el arbol podado más antiguo de A y B

```
1: Antiguo(RA_i,RB_j)
2: if (RA_i.t_e > RB_j.t_e) then
3:    j++;
4: else
5:    i++;
6: end if
```

El **algoritmo 2** procesa exactamente aquellos 3DR-tree-podados que se intersectan temporalmente, cuyo costo se reduce por la restricción de intersección espacial que pueda existir entre pares de 3DR-tree-podados.

4. Modelo de costo

Asumimos que todos los objetos están uniformemente distribuidos en un espacio d-dimensional[18]. En segunda instancia, la historia T de la base de datos se ha particionado en K subestructuras de altura H_i, obteniéndose K 3DRp-tree (3DR-tree podado, en adelante árbol). Los que son responsables de un intevalo de vida (*lifespan*) promedio $L_i = \sum_{i=1}^{K} L_i/K$

Sea $N_{R_i,j}$ el número de nodos del árbol i en el j-ésimo nivel, y sea $P_{i,j}$ la probabilidad de que un nodo del árbol i, a nivel j sea visitado para responder la consulta de reunión. Suponiendo que los tamaños promedios espacial y temporales de los nodos de los árboles son iguales. Entonces la cantidad de nodos accesados para responder una consulta de reunión espacio-temporal que involucre un único árbol de A y un único objeto del conjunto B estará dada por la Ec(3).

4.1 Costo de la reunión mejorada

De acuerdo a estas consideraciones aplicadas al algoritmo de reunión espacio-temporal mejorado sobre los árboles de los conjuntos A y B tenemos:

Cuadro 1: Lista de símbolos usados

Símbolo	Definición
M	capacidad máxima del nodo
f	fanout promedio del nodo
c	capacidad promedio de nodo en %
h	altura del arbol primario 3DR-tree .
H	altura promedio del subarbol podado 3DRp.
N_i	Número de objetos en el conjunto inicial de datos
$N_{i,j}$	Número de entradas del arbol i en el nivel j.
D_i	Densidad de nodo a nivel i
$S1$,S_2	Conjuntos de datos 1 y 2
a_i	Agilidad del árbol i
T	Número total de instantes de tiempo en la historia de la base
L_i	amplitud del intervalo de vida de un árbol podado;
$s_{i,j}$	Tamaño promedio del nodo s en el arbol i en el nivel j
NA_i	Número promedio de nodos accesados por la operación en el arbol i
NA	Número total promedio de nodos accesados en la reunión
$C(STJ)_{3DRp}$	Costo de reunión espacio temporal sobre 3DR podado.
K	Número de arboles podados .

$$C_{STJ_{3DR1}} = [\mathbf{C(A)} + \mathbf{C(B)}] \cdot \mathbf{P}^{'} \cdot \mathbf{C(SJ)_{3DRp}}$$

donde C(A)=K1 y C(B)=K2 son el costo leer los conjuntos A y B respectivamente y $\mathbf{P}^{'}$ es la probabilidad de intersección espacial de los 3DR-tree-podados de A y B. En este caso se trata de hallar la probabilidad de intersección del MBR que cubre todo el 3DR-tree-podado de uno de los árboles del conjunto A, con el MBR que cubre todo un 3DR-tree-podado del conjunto B. Observamos que el área máxima que determinan estos dos 3DR-tree-podados es igual a la suma de sus respectivos MBR ($RA_i.MBR + RB_j.MBR$), siendo el universo muestral el área ocupada por el MBR de todo el conjunto B y cuya expresión esta dada por S2.MBR $= \sum_{j=1}^{K2} \cdot \sum_{n=1}^{H-1} N2_{j,n} \cdot s2_{j,n}$, donde $N2_{j,n}$ es la cantidad promedio de nodos del j-ésimo árbol podado del conjunto B a nivel n y $s2_{j,n}$ es el tamaño promedio del nodo de ese árbol a nivel n. Relacionando ambas expresiones a través del concepto de probabilidad, determinamos que la probabilidad espacial de intersección de los árboles podados RA_i y RB_j esta dada por :

$$P_\theta = \frac{RA_i.MBR + RB_j.MBR}{K2 \cdot \sum_{n=1}^{H-1} N2_n \cdot s2_n}$$

$$\text{donde : } RA_i.MBR = \sum_{n=1}^{H-1} N1_{i,n} \cdot s1_{i,n} \qquad \text{y} \qquad RB_j.MBR = \sum_{n=1}^{H-1} N2_{j,n} \cdot s2_{j,n}$$

$s1_{i,n}$ es el tamaño promedio de los nodos del i-ésimo 3DR-tree-podado del conjunto $A \in S1$ a nivel n y $s2_{j,n}$ es el tamaño promedio de los nodos del j-ésimo 3DR-tree-podado del conjunto $B \in S2$ a nivel n y cuyos valores se obtienen de la ecuación: $s_{i,n} = \sqrt{D_{i,n}/N_{i,n}}$ [19]. En la ecuación utilizaremos la expresión para la probabilidad $P^{'} = min[1; P]$ dada por [14].
Luego reemplazando valores nos queda:

$$\Longrightarrow \mathbf{C_{(STJ1)_{3DRp}}} = [K1 + K2] \cdot min[1; P_\theta] \cdot C(SJ) \tag{5}$$

$C(SJ)$ es el costo de la reunión de los elementos de las estructuras de RA_i y RB_j que pertenecen a los arreglos A y B y que satisfacen el predicado de reunión espacio-temporal.

La expresión para el costo $C(SJ)$ de acuerdo a la Ec(3) considerando el intervalo temporal de los árboles podados, vale:

$$\mathbf{C(SJ)_{3DRp}} = \sum_{n=1}^{H_j-1} N_{RB_{j,n}} \cdot \left\{ N_{RA_{i,n}} \cdot P_{i,n} + 1 + N_{RA_{i,n+1}} \cdot [\ P_{i,n+1}]^2 \right\} \quad (6)$$

donde $N_{RA_{i,n}} = [N_{R_i,n} + N_{R_i,n}(L_i - 1) \cdot a_*]$ es el número de nodos del árbol i al cabo de un período de tiempo L_i (lifespan del árbol i) a nivel n, y $N_{R_i,n} = (N_{i,n}/f)$ es la cantidad inicial de nodos en el i-ésimo árbol podado a nivel n, donde N_i es la cantidad inicial de objetos del árbol podado, cuya expresión esta dada por :

$$N_i = \frac{(N_t/K)}{(L_i - 1) \cdot a\,\% + 1}$$

donde N_t es la cantidad de objetos del 3DR-tree sin podar al cabo de la historia T; con lo cual la cantidad de objetos en un árbol podado al cabo de L_i instantes de tiempo esta dada por :

$$N_{L_i} = N_i + N_i \cdot (L_i - 1) \cdot a_* \quad (7)$$

5 Evaluación.

En esta sección comparamos la eficacia del modelo propuesto con el modelo 3DR-tree. La razón de usar 3DR-tree es que es uno de los métodos más eficientes para procesar consultas considerando intervalos temporales largos teniendo un rendimiento muy similar a MV3R-tree [15]. Consideramos conjuntos de hasta 50.000 objetos de tipo punto los que se distribuyeron uniformemente en el espacio. Consideramos una historia T1 de 100 instantes de tiempo y para T2 de 100 y 200 instantes de tiempo. La unidad de medida utilizada correspondió a la cantidad de bloques de disco (1024 bytes) leídos al ejecutar la operación de reunión espacio temporal.

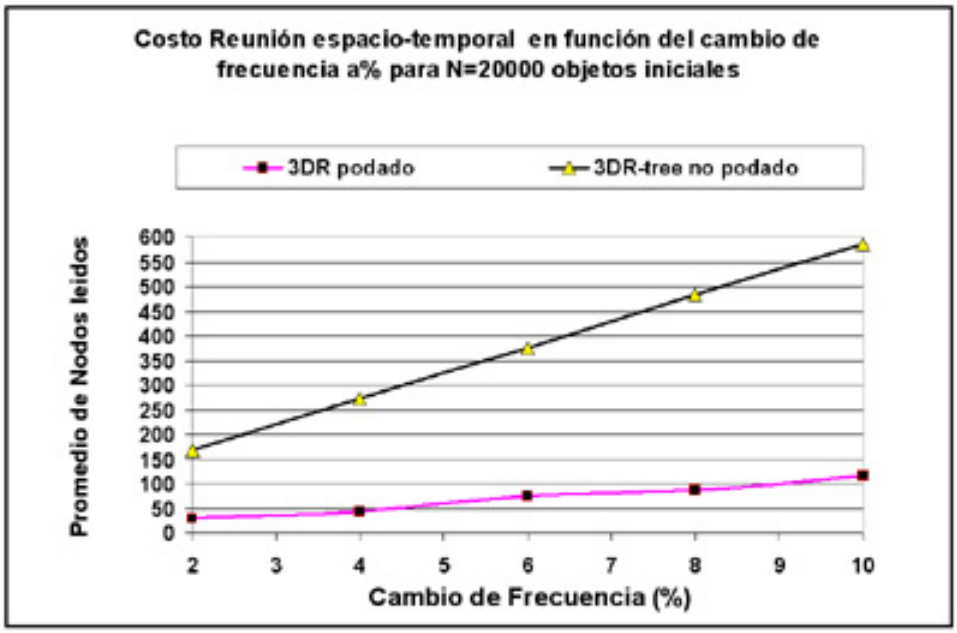


Figura 2: Costo en función del cambio de frecuencia

5.1 Comparación analítica

El rendimiento de 3DR-tree se obtuvo mediante el modelo de costo descrito en la Sección 2.3 Ec.(3).

El total de objetos a insertar en el 3DR-tree se obtuvo con la Ec.(4), y el total de objetos a insertar en los 3DR-tree-podados se obtuvo con la Ec.(7). Por otro lado consideramos una densidad inicial $D_0 = 0{,}2$, con una capacidad por nodo de 36 entradas y una capacidad promedio de llenado de 67 % . Se adoptó un valor K1=10 para la partición del la estructura principal de cada conjunto de datos y se obtuvo un intervalo temporal promedio por árbol podado de $L_i = 10\,\%$ de T . Para el conjunto S2 adoptamos un rango de K2=(5,10,20,40) obteniendo un rango de intervalos temporales promedios para cada valor de K2. Por otro lado tambien se probaron extensiones temporales en un rango de variación desde T1=100 unidades temporales, y T2=100 y 200 unidades temporales.

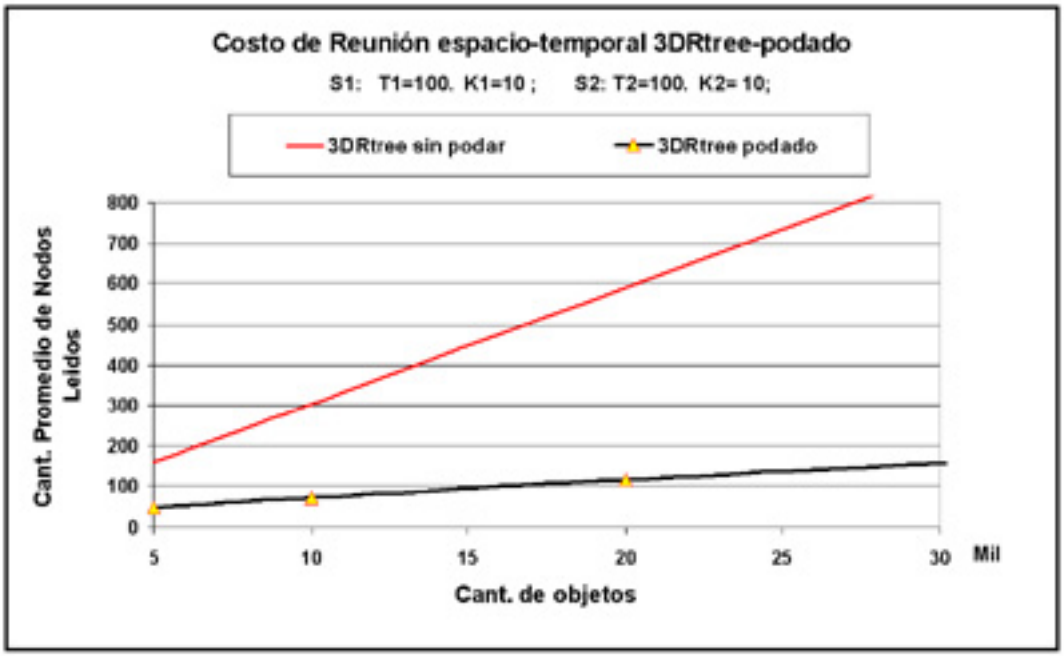


Figura 3: Costo de la reunión espacio-temporal

El gráfico de la Figura 2 se ha obtenido mediante la implementación de 3DR-tree y APR-tree. En dicha figura es posible observar el rendimiento del modelo para la reunión espacio-temporal sobre las estructuras podadas respecto de la consulta sobre una estructura 3DR-tree sin podar. Los resultados están en función del *cambio de frecuencia a %*, normalmente llamado movilidad, la cual representa el porcentaje de objetos que cambian su posición en cada instante de tiempo. La diferencia observable se debe fundamentalmente a la reducción en altura de los árboles respecto de la altura del árbol principal y la reducción del orden de complejidad de $O(K1 \cdot K2)$del *Algoritmo 1* a $O(K1 + K2)$ dada por el *Algoritmo 2*. Esta característica se ve reflejada en el tamaño promedio del nodo que al reducir la altura y la cantidad de objetos de búsqueda por árbol, la probabilidad de intersección entre los nodos crece en proporción al tamaño de los mismos, lo cual redunda en una mayor exactitud en el filtrado de las extensiones espaciales.

En la Figura 3 se puede observar el efecto de mejora que se obtiene al particionar la estructura 3DR-tree principal en diferentes cantidades de intervalos temporales pero manteniendo igual rango temporal para ambos conjuntos de datos e igual cantidad de particiones K. En cambio cuando aumentamos el rango temporal para un conjunto y variamos la cantidad de particiones, el costo es menor para valores de $K2 > K1$. Esto se observa en la Figura 4. Esta diferencia se basa en que los intervalos temporales L_i para

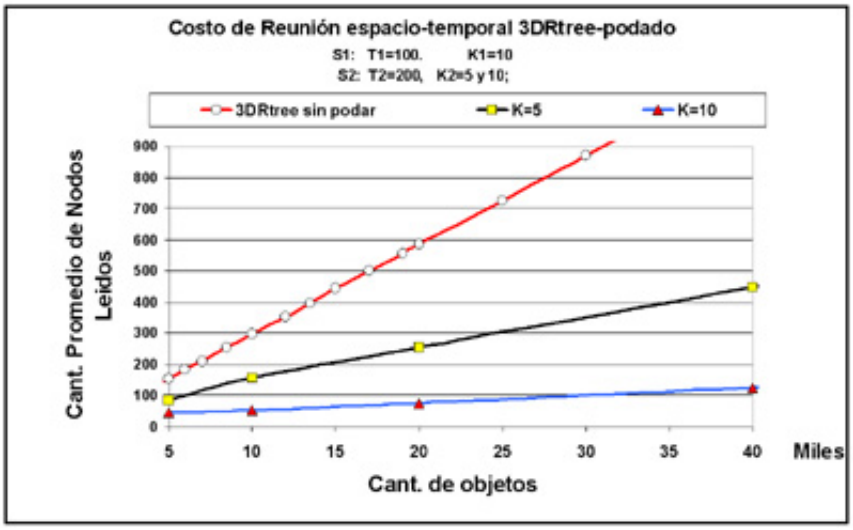


Figura 4: Costo para diferentes valores de K2 y distinto T

valores pequeños de $K2$, aumentan y este aumento genera una gran cantidad de entradas en los árboles podados lo que redunda en un aumento del costo.

6. Conclusiones.

En este trabajo se presenta un algoritmo para realizar la reunión espacio-temporal basado en la estructura 3DR-tree podada (APR-tree). La aplicación de la metodología de poda permite reducir la altura del árbol de búsqueda y por lo tanto reducir el costo de las operaciones de consulta que se realizan. Se definió un modelo de costo para nuestro algoritmo. Usando los modelos de costo de 3DR-tree y el de nuestro algoritmo, se compararon los rendimiento de la reunión espacio-temporal usando ambas estructuras.

Como se ha podido observar el costo de la reunión espacio temporal sobre los árboles podados es bastante menor que la observada para toda la estructura sin podar.

El siguiente paso en este trabajo es comparar la eficacia del modelo descrito, implementando el algoritmo usando el lenguaje c++ sobre una plataforma linux. Una vez realizado esto se determinará cuan ajustado a la realidad se comporta el modelo establecido para ambos algoritmos de reunión (join) sobre 3DR-tree.

Como trabajo futuro se pretende ahondar en el estudio de los algoritmos de reunión (join) espacio-temporales, en particular para consultas del tipo "*Hallar las colisiones de autos que ocurrieron el dia de ayer en la zona céntrica*". Este tipo de consulta evidentemente se responde con la evaluación de la operación de reunión espacio-temporal pero se ha de puntualizar que ahora la consulta especifica un sector determinado donde se consultan las intersecciones y que permite realizar un corte más específico en este tipo de consultas. Tambien enfocaremos el diseño de estructuras de datos y algoritmos que permitan resolver eficientemente este tipo de consultas y realizar trabajos de experimentación que permitan establecer la eficacia de tales modelos comparándolos contra los métodos existentes.

Referencias

[1] Brinkhoff, T., Kriegel, H.-P., and Seeger, B.Eficient processing of spatial joins using R-trees. In ACM SIGMOD Conference on Management of Data (Washington, DC, USA, 1993), ACM, pp. 237-246.

[2] Gutiérrez, G., Navarro, G., and Rodríguez, A Spatio-temporal Access Method based on Snapshots and Events.(GIS'05), November 4,2005, Bremen, Germany. Copyright 2005 ACM 1-59593-146-5/05/0011

[3] Gutiérrez, G., Navarro, G., and Rodríguez, A. SestL: An event-oriented spatio-temporal access method. Tech. Rep. TR/DCC-2006-5, Department of Computer Science, Universidad de Chile. 2006.

[4] Guttman, A. R-trees: A dynamic index structure for spatial searchinng. In ACM SIGMOD Conference on Management of Data (1984), ACM,pp. 47-57.

[5] Hadjieleftheriou, M.,Kollios,G., Tsotras, V.J. and Gunopoulos,D. Efficient indexing of spatiotemporal objects. In Extending Database Technology (2002), pp.251-268.

[6] Huang,Yun-Wu. Ning Jing, Elke A. Rundensteiner: Spatial Joins Using R-trees: Breadth-First Traversal with Global Optimizations. VLDB 1997: 396-405

[7] Hyung-ju Cho, Jun-Ki Min, Chin-Wan Chung. And Adaptive Indexing Technique Using Spatio-Temporal Query Workloads. Korean Advanced Institute of Science. Information and Software Technology, Vol 46, pp.229-241 March 2004.

[8] Kollios, G. N. Indexing Problems in Spatiotemporal Databases. PhD thesis, Polytechnic University, New York, June 2000.

[9] Nascimento,M., Silva, J., and Theodoridis, Y. Access structures for moving points. Tech. Rep.. TR-33, TIME CENTER, 1998.

[10] Pagel, B.U., Six,H., Toben, H. and Widmayer P. Towards an analysis of range query performance in spatial data structures. In PODS'93: ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Databases Systems (New York, USA 1993).ACM Press,pp.214-221.

[11] Seung-Hyun, J., Paton, W., Fernandes, A. A., and Tony, G. An experimental performance evaluation of spatio-temporal join strategies. Transactions in GIS (March 2005),129-156.

[12] Tao, Y., and Papadias, D. Eficient historical R-Tree. In SSDBM International Conference on Scientific and Statical Database Management (2001), pp. 223-232.

[13] Tao, Y., Papadias, D., and Zhang, J. Cost models for overlapping and multiversion structures. ACM Trans. Database Syst. 27, 3 (2002), 299-342.

[14] Tao, Y., Papadias, D. Historical Spatio-Temporal Aggregation. ACM Transactions on Database Systems, Vol.23, No.1, January 2005,pp:61-102.

[15] Tao, Y., Papadias, D. MV3R-tree: A spatio-temporal access method for timestamp and interval queries. In The VLDB *Journal* (2001), pp. 431-440.

[16] Theodoridis, Y., and Sellis, T. A model for the prediction of R-tree performance. In PODS '96: Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (New York, NY, USA,1996), ACM Press, pp. 161-171.

[17] Theodoridis, Y.,Vazirgiannis, M. and Sellis, T. Spatio-Temporal Indexing for Large Multimedia Applications. In Proc. of the IEEE Conference on Multimedia Computing and systems, ICMCS, june 1996.

[18] Theodoridis, Y., Stefanakis, E., and Sellis, T. Eficient cost models for spatial queries using R-Trees. IEEE Transactions on Knowledge and Data Engineering 12, 1 (2000), 19-32.

[19] Theodoridis, Y., Stefanakis, E., and Sellis, T. An Eficient cost Model for spatial Joins Using R-Trees. IEEE Transactions on Knowledge and Data Engineering (1997).

[20] Varzirgiannis M., Theodoridis, Y., Sellis, T. Spatio-Temporal Composition and indexing for Large Multimedia Applications. ACM Multimedia Systems, 6(5), 1998.

AMBIENTE DE EXPLOTACION DE INFORMACIÓN BASADO EN LA INTEGRACION DE CLASIFICACIÓN, SELECCIÓN Y PONDERACIÓN DE REGLAS

G. Schulz[1], E. Fernández[1,2], H. Merlino[1,2], D. Rodríguez[2], P. Britos[2,1], R. García-Martínez[2,1]

[1]Laboratorio de Sistemas Inteligentes, Facultad de Ingeniería, Universidad de Buenos Aires.
Paseo Colón 850 4to Piso. Ala Sur.
(1063) Capital Federal, ARGENTINA.
[2]Centro de Ingeniería de Software Ingeniería del Conocimiento. Escuela de Postgrado. ITBA
25 de Mayo 444 – 6to. Piso
Capital Federal, República Argentina
rgm@itba.edu.ar

Resumen: Actualmente no existe un escenario que integre las funciones de clasificación de instancias, selección y ponderación de reglas, y por lo tanto utilizar a cada una de estas funciones como complemento uno del otro, para lograr una profunda y completa investigación de las características de las poblaciones que se desean estudiar. Esta falencia hace que cada vez que se quiera, por ejemplo, extraer las reglas de producción que dan como consecuencia la clasificación de una población, se necesite primero clasificar a los individuos de una población en un escenario de clasificación, para luego ingresar a estos individuos clasificados en un escenario diferente, capaz de inferir y extraer las reglas. Aquí se propone desarrollar un ambiente capaz de integrar las tres funciones.

Palabras Clave: inducción de reglas, clasificación automática, elección automática de reglas, integración de inducción y ponderación.

Abstract: At the moment it does not exist a scene that integrates the mechanisms of classification of instances, selection and ponderación of rules, and therefore to use to each one of these mechanisms as complement one of the other, to obtain a deep and complete investigation of the characteristics of the populations that are desired to study. This falencia does that whenever it is wanted, for example, to extract the production rules that give like consequence the classification of a population, is needed first to classify to the individuals of a population in a classification scene, soon to enter these individuals classified in a scene different, able to infer and to extract the rules. Here one sets out to develop a tool able to integrate the three mechanisms.

Key words: rule induction, machine classification, rule selection, induction and weighting integration.

1. INTRODUCCIÓN

Existen numerosos ambientes que utilizadas en forma exitosa tanto para clasificar a una población de individuos, para inferir reglas inherentes a las características de una población o para ponderar reglas. Sistemas que utilizan a las redes neuronales son un ejemplo de eso, ya que dependiendo de la arquitectura de redes que utilicen, se comportan muy bien como clasificadores de elementos de un dominio; los sistemas que implementan árboles de decisión tales como ID3 [1] o C4.5 [2], por otro lado, son también muy comunes en lo que se refiere a la extracción de reglas de dominios o que utilizan a las redes Bayesianas como modelos de ponderación de reglas
En la tabla 1 se relacionan varios de los softwares actualmente disponibles en el mercado, junto con una pequeña reseña de las funciones que proveen y de las técnicas utilizadas para brindar esas características.

AMBIENTE	**DESCRICION**
AC^2	*AC^2* es un ambiente de data mining diseñada para usuarios conocedores de la materia. *AC^2* tiene un modelado grafico orientado a objetos y librerías en C y C++. Soporta la edición interactiva del árbol que se genera. Se comporta como una librería multiplataforma de funciones de data mining. Provee como funciones: clusterización, clasificación, predicción, segmentación. Utiliza como técnica árboles de decisión [3].
AnswerTree	*AnswerTree* es un ambiente de SPSS utilizado para construir árboles de decisión. Como ambiente de data mining apunta perfilar a grupos para la comercialización y las ventas. Utiliza cuatro algoritmos de árboles de decision. Incluidos están dos algoritmos CHAID, los cuales SPSS ha extendido para manejar categorización nominal, ordinal y variables continuas dependientes. Provee como funciones: Clasificación. Utiliza como técnicas: Árboles de decisión (CHAID, CHAID Exhaustivo, C&RT (variación de CART), QUEST). [4]
CART	*CART* es un ambiente de árbol de decisión que utiliza el algoritmo CART. Para poder manejar la falta de información, los datos son manejados a través de reglas de backup que no siempre asumen que todos los datos de un atributo incierto es el mismo. Se utilizan siete criterios diferentes de splitting (incluyendo el Gini). Debido al uso del motor de traducción de datos, *DBMS/Copy*, se pueden utilizar datos de diferentes tipos de formato (incluyendo Excel, Informix, Lotus, Oracle). Provee como funciones: Clasificación. Utiliza como técnicas: Árboles de decisión (CART). [5], [6].
Clementine	*Clementine* utiliza iconos descriptivos como interfaz, el usuario crea descripciones de flujos de datos de las funciones que se realizarán. Cada icono representa un paso en el proceso total de minería de datos. Existen incluidos iconos para funciones tales como el acceso a datos, preparación de datos, visualización y modelado. Para asistir a la creación de secuencias, Clementine utiliza Capri. Ademas puede utilizar grandes conjuntos de datos usando un modelo de cliente/ servidor. Cuando es posible, el servidor convierte peticiones del acceso a los datos en las consultas SQL, que pueden entonces tener acceso a una base de datos emparentada. Provee como funciones: Reglas de asociación, clasificación, clusterización, análisis de factor, pronostico, predicción. Utiliza como técnicas: Apriori, BIRCH, CARMA, árboles de decisión (C5.0, C&RT variación de CART),

AMBIENTE	DESCRICION
	clusterización K-means, redes neuronales (Kohonen, MLP, RBFN), regresión (lineal, logística) inducción de reglas (C5.0, GRI). [7]
Weka	*Weka* contiene se focaliza en algoritmos de clasificación, regresión, y clusterización de patrones. Weka es un software gratuito y open-source bajo la licencia al público en general del GNU (GLP). Las técnicas que utiliza son: Naïve Bayes, Nearest neighbor, Linear models, OneR, Decision trees, Covering rules, K-means, EM, Cobweb. [8]

Tabla 1. Ambientes de minería de datos disponibles en el mercado

2. PROBLEMA A RESOLVER

El problema o la falencia de los ambientes anteriormente detallados es que ninguno de ellos logra integrar y complementar las tres funciones en su implementación. Esto hace que cada vez que se quiera, por ejemplo, extraer las reglas que dan como consecuencia la clasificación de una población, se necesite primero clasificar a los individuos de una población en un escenario de clasificación X, para luego ingresar a estos individuos clasificados en un escenario diferente, capaz de inferir y extraer las reglas. Lo mismo ocurriría si se necesita ponderar estas reglas obtenidas. En la Figura 1 se muestra un posible escenario de lo arriba planteado. Allí se observan que son necesarios tres escenarios para poder extraer las reglas inferidas de clasificación.

1) *Escenario de Clasificación*: Recibe como entrada los datos a clasificar. Su función es la de clasificar a esos datos. La salida da como resultado los datos clasificados, en formato A.
2) *Escenario de Transformación de Datos:* Recibe como entrada datos clasificados en un formato A. Su función será la de transformar esos datos que están en formato A al formato B, para que sean entendidos por el escenario 3.
3) *Escenario de Selección de Reglas:* Recibe como entrada los datos clasificados en formato B. Su función es la de inferir las reglas que dieron origen a la clasificación. Su salida son el conjunto de reglas inferidas en formato B.

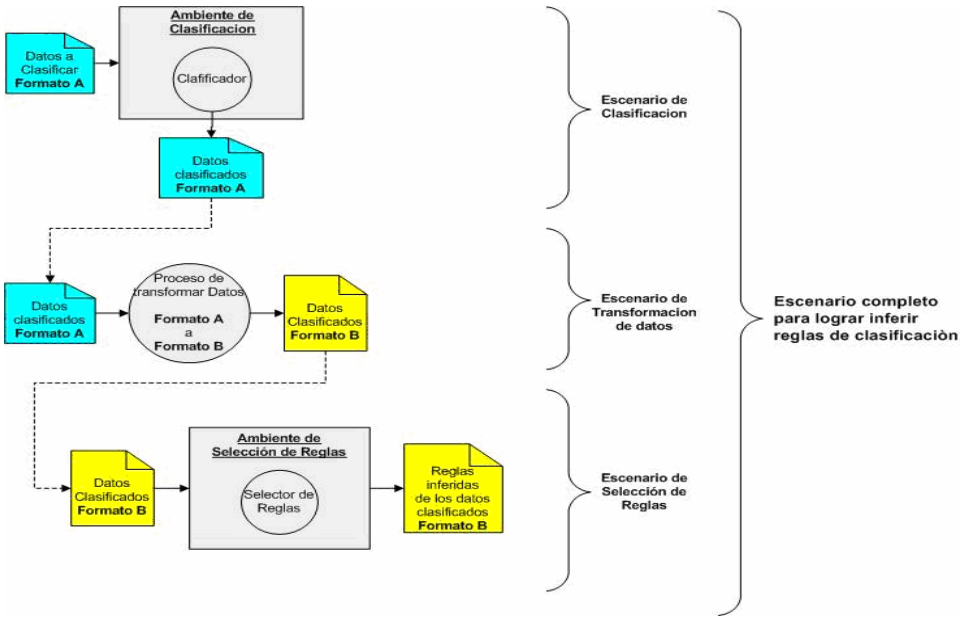


Figura 1. Tres posibles escenarios para inferir reglas de clasificación.

3. SOLUCIÓN

Lo que se plantea con este trabajo es desarrollar un ambiente que provea las funciones de:

- clasificar a una población.
- inferir las reglas que determinan que un determinado individuo pertenezca a cierta clase
- determinar la probabilidad de ocurrencia de una regla (ponderar).

Además esas capacidades, el ambiente va a tener que ser capaz de integrar y complementar a cada una de ellas, logrando que cada una sea el complemento de la otra. Cada uno de las funciones complementará su trabajo con el trabajo de las otras funciones. Así va a ser posible clasificar a una población, luego extraer o inferir las reglas que dieron origen a dicha clasificación para finalmente ponderar dichas reglas, para obtener la probabilidad de ocurrencia de cada una de las reglas antes inferidas. Para lograr implementar cada uno de estos funciones, se van a utilizar redes neuronales denominadas *mapas autoorganizados* [9] para la clasificación, mediante árboles de decisión como lo son los *ID3* [1] se buscará inferir las reglas de clasificación, y se van a utilizar mecanismos probabilísticos, como lo son las *Redes Bayesianas*, para determinar la probabilidad de ocurrencia de una regla. Para lograr la integración de estas funciones, en la Figura 2 se muestra el flujo de información existente dentro del ambiente. En la figura se observan varios procesos que conforman este flujo. En la Tabla 2 se detalla cada proceso:

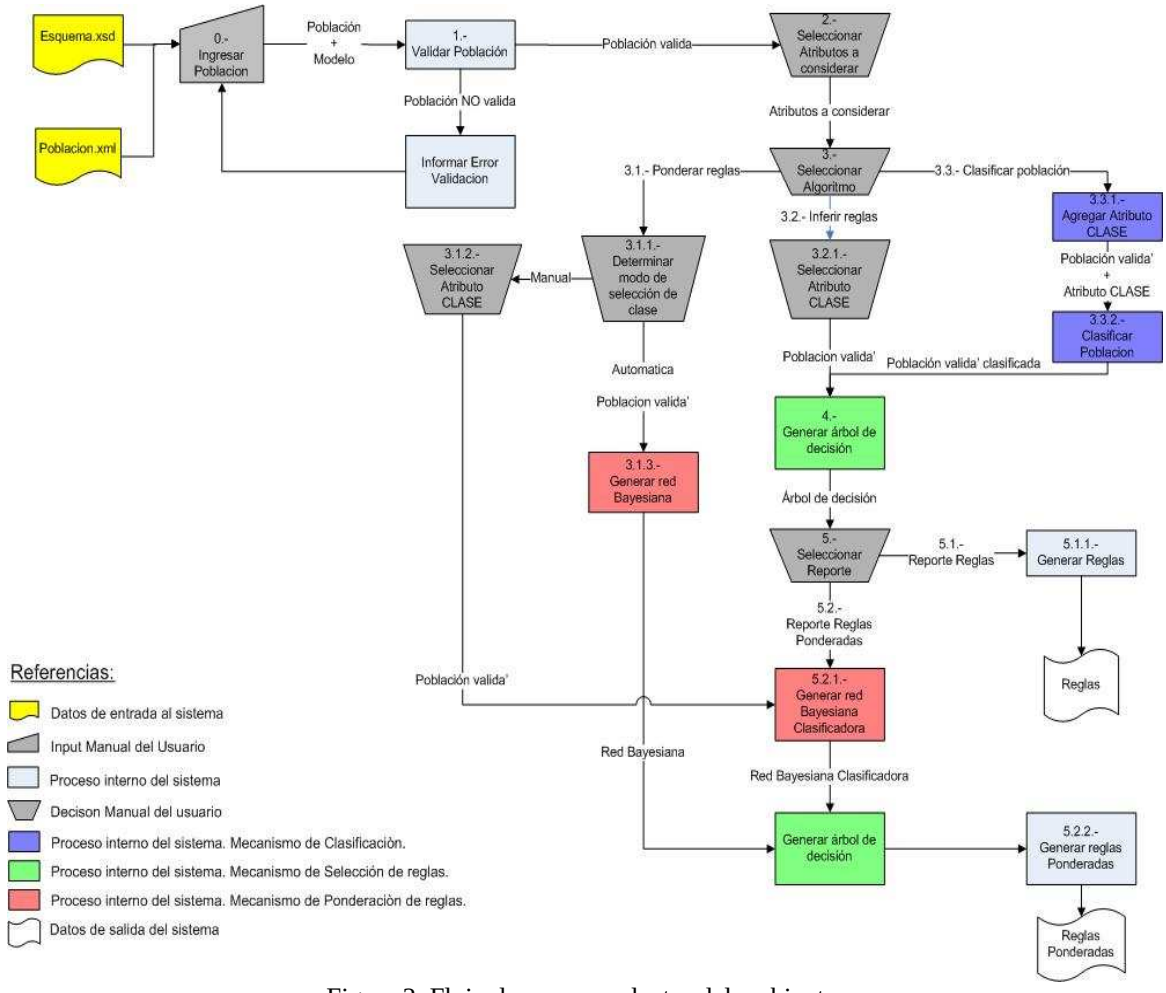


Figura 2. Flujo de procesos dentro del ambiente.

PROCESO	PROCEDIMIENTO
0. Ingresar población	El usuario se limita a determinar cual va a ser el archivo que representa a la población que se va a procesar. Una vez que el usuario selecciona el archivo, el ambiente automáticamente intenta buscar dentro del mismo directorio donde se encuentra este archivo población, el archivo esquema que representa el modelado de esta población. Para ello busca un archivo con el mismo nombre que el de la población, pero con extensión xsd en lugar de xml. Si no encuentra al archivo esquema, entonces produce un error informando de la no existencia de este archivo. Si el ambiente encuentra el archivo xsd, entonces se procesan cada uno de los dos archivos, guardándose en distintas instancias de objetos los datos característicos de cada uno de estos archivos. A cada una de estas instancias llamaremos *Población* y *Modelo* respectivamente.
1. Validar población	En esta etapa se realiza la validación del dominio o población que ingresó al ambiente. Para ello lo primero que se hace es verificar que cada uno de los individuos de la población defina los atributos especificados en el *Modelo*, y que los tipos de datos de estos atributos sean válidos de acuerdo a lo que especifica este *Modelo*. Una vez que se comprueba que lo anterior es correcto, se recorre uno a uno los individuos del objeto Población, y dentro de cada individuo se evalúa que cada uno de los atributos que lo caracterizan tenga un valor valido, de acuerdo a lo que se especifica en *Modelo*. En caso de encontrase alguna inconsistencia en los datos de la población, el sistema informará mediante un mensaje el motivo por el cual no se pudo realizar la validación.
2. Seleccionar atributos a considerar.	El usuario selecciona, del total de atributos que caracterizan a la población, un subconjunto de estos atributos con los cuales desea que se realice el análisis de la población en estudio. Estos atributos son los que se considerarán de ahora en más en todo el proceso, y determinarán a lo que llamaremos Población valida'. Básicamente esta Población valida' estará compuesta de los mismos individuos que la Población válida, solamente que estos individuos serán determinados por un subconjunto de atributos, y no necesariamente por el total. Supongamos que los atributos Edad, Peso y Altura son los atributos que caracterizan a la población y el usuario elige como atributos a considerar sólo Edad y Altura. De esta manera, lo que llamamos Población valida' serán individuos determinados solamente por estos dos atributos.
3. Seleccionar algoritmo	Este es uno de los procesos donde la decisión del usuario es fundamental para la continuación del flujo y procesos del ambiente. Aquí el usuario decide que algoritmo va a utilizar para continuar con el estudio de la población. Los posibles algoritmos a elegir son los siguientes: Ponderación de reglas, Inferir reglas o Clasificar población.
3.1. Ponderación de reglas	Al elegir este algoritmo, el usuario está determinando que el único proceso que necesita realizarle a la población es la ponderación de reglas de decisión que dieron origen a la clasificación de la población. Obviamente al elegir este algoritmo, se presupone que la población en estudio es una población que ingreso al mismo ya clasificada. El usuario va a tener la opción de elegir que atributo es el que determina el atributo clase, o si va a ser la propia del ambiente la encargada de seleccionar este atributo clase. Una vez que se genere la red Bayesiana, para poder determinar cuales son las reglas a inferir, va a ser necesario que el ambiente genere el árbol de decisión. Este punto es transparente al usuario, pero necesario para poder determinar cuales son las reglas.
3.1.1. Determinar modo de selección de clase	El usuario determina si la selección del atributo clase para el procesamiento de la red Bayesiana la deberá hacer automáticamente el sistema, o va a ser el propio usuario el que determinará cual de los atributos que conforman a la población valida' será el atributo clase. Si la selección del atributo clase la deberá hacer el sistema, entonces la red Bayesiana que va a generar el sistema va a ser una red Bayesiana tradicional, y el propio proceso de generación de esta red determinará, como consecuencia de este proceso, cual es este atributo clase. En cambio, si es el usuario quien selecciona qué atributo es el denominado atributo clase, entonces la red Bayesiana que se generará será una red Bayesiana de clasificación.

Tabla 2. Procesos propuestos para el ambiente

3.1.2. Seleccionar atributo clase	Si en el proceso 3.1.1 el usuario selecciono que manualmente iba a determinar que atributo sería en atributo clase, en este proceso deberá seleccionar del subconjunto de atributos que caracterizan a la población valida' cual de ellos es el atributo clase. A partir de esta elección, el sistema deberá generar una red Bayesiana de clasificación, cuyo atributo clase es precisamente el atributo seleccionado por el usuario.
3.1.3. Generar red bayesiana	El sistema genera una red bayesiana, valiéndose de la población valida' como datos de entrada para el proceso de entrenamiento y testeo de la red que generará. Será el ambiente, luego de generada la red Bayesiana, el que deberá determinar, según las características de la red que genere, que atributo se determinó como atributo clase.
3.2. Inferir reglas	Al elegir este algoritmo, la lectura que debemos hacer es que la población que ha ingresado al ambiente es una población ya clasificada, por lo que la necesidad del usuario recae en lograr información sobre aspectos que no tienen que ver con una clusterización de la población, sino con la de lograr determinar las reglas de decisión que dieron por origen la clasificación de esos individuos
3.2.1. Seleccionar atributo clase	Como la población que se ha ingresado al ambiente es ya una población clasificada, hay que definirle al ambiente cual de todos los atributos a considerar de la población es el que determina a que clase pertenece cada individuo. Esto lo determina el usuario.
3.3. Clasificar la población	Al elegir este algoritmo, lo que está planteando el usuario es una necesidad de clusterizar primero a la población, entendiéndose con esto que la población no tiene determinado ningún atributo que describa a que clase pertenece cada individuo. Esto significa que será el ambiente el encargado de realizar esta tarea, y lo hará mediante un algoritmo que no necesita de ninguna supervisión, por lo que el ambiente asume la total responsabilidad de la tarea de clusterizar a la población.
3.3.1. Agregar atributo clase	Como la necesidad del usuario es de clusterizar a la población, en este proceso es el sistema el que agrega un nuevo atributo, denominado CLASE, al conjunto ya existente de atributos característicos de la población. El valor que tome este nuevo atributo es el que determinara a que clase pertenecerá cada uno de los individuos, una vez realizada la clasificación.
3.3.2. Clasificar población	En este proceso se clusteriza a la población, determinándose el valor que tomará, para cada uno de los individuos, el atributo CLASE. La cantidad de clases en la que el ambiente intentará clasificar a los individuos es un valor que el propio usuario del ambiente determinará. La forma con la que se implementa este proceso de clusterización es mediante la utilización redes neuronales denominadas de aprendizaje competitivo y cooperativo. Con este tipo de aprendizaje se pretende que cuando se presente a la red cierta información de entrada, solo una de las neuronas de salida de la red se active o alcance su valor de respuesta máximo. Es por eso que las neuronas compiten para activarse, quedando finalmente una como neurona ganadora, mientras que el resto quedan anuladas. Los individuos con características similares son clasificados formando parte de la misma categoría y por lo tanto deben activar la misma neurona de salida.
4. Generar árbol de decisión	Las instancias del dominio o población con las clases a las que pertenecen son presentadas al ambiente, quien como consecuencia de realizar la tarea de inducción, generará un árbol de decisión.
5. Seleccionar Reporte	El usuario simplemente elige que tipo de reporte quiere obtener del ambiente. Puede optar por el Reporte de Reglas, donde el ambiente solamente presentará las reglas que dieron origen a la clasificación, o puede elegir el Reporte de Reglas Ponderadas, donde el ambiente además de presentar las reglas, también determinará la probabilidad de ocurrencia para cada una de esas reglas.

Tabla 2. Procesos propuestos para el ambiente (cont.)

5.1. Reporte de Reglas	
5.1.1. Generar Reglas	El árbol de decisión es recorrido desde la raíz hasta cada una de las hojas, y se generarán las reglas de decisión interpretando o mapeando cada bifurcación del árbol con su respectivo atributo y valor que la bifurcación tome. Las reglas generadas serán del estilo *SI Atributo1 = valor1 Y* *Atributo2 = valor2 Y* *… Y* *AtributoN = valorN* *ENTONCES* *Clase = clase1.*
5.2. Reporte de Reglas Ponderadas	
5.2.1. Generar red Bayesiana clasificadora	Las instancias del dominio o población clasificadas son presentadas al ambiente. El ambiente utiliza estas instancias como datos de entrenamiento para generar, mediante un algoritmo de entrenamiento supervisado, de una red Bayesiana clasificadora. El tipo de algoritmo a utilizar para el entrenamiento de la red bayesiana es un dato que lo determina el usuario del ambiente.
5.2.1. Generar reglas ponderadas	El árbol de decisión generado a partir de las instancias clasificadas es recorrido por el procesador de reglas, el cual generará las reglas de decisión interpretando o mapeando cada bifurcación del árbol con su respectivo atributo y valor que éste tome en la bifurcación. Para cada una de estas reglas, utilizará a la red bayesiana para poder determinar la probabilidad de ocurrencia de esta regla. Las reglas generadas serán del estilo *SI Atributo1 = valor1 Y* *Atributo2 = valor2 Y* *… Y* *AtributoN = valorN* *ENTONCES* *Clase = clase1.* *PROBABILIDAD* *% de probabilidad de ocurrencia de la regla inferida*

Tabla 2. Procesos propuestos para el ambiente (cont.)

4. CASOS DE ESTUDIO

Para comprobar experimentalmente el correcto funcionamiento del ambiente, se analizó su uso en diferentes casos de estudio. Para realizar esta experimentación, se utilizaron bases de datos obtenidas del *UCI Machine Learning Repository* del Departamento de Información y Ciencias de la Computación de la Universidad de California. A continuación se resumen las características de la base de datos utilizadas en la tabla 3.

Base de datos	# Atributos	# Instancias	Descripción de la base de datos
Zoología	18	101	Una base de datos simple que contiene 16 atributos booleanos y uno numérico que definen diferentes animales. El atributo "tipo" define el atributo clase.

Tabla 3. Bases de datos utilizados

Se realizó la clasificación de instancias, especificando para que se clasifique en 7 clases diferentes. En la Tabla 4 y en el grafico de la Figura 3 se observan los resultados obtenidos.

Clase	Individuos
Clase0	aardvark, antílope, oso, jabalí, búfalo, becerro, cobayo, chita, venado, elefante, jirafa, girl, cabra, gorila, hámster, liebre, leopardo, león, lince, visón, topo, mangosta, zarigüeya, orix, ornitorrinco, turón, pony, puma, minino, mapache, reno, foca, lobo marino, ardilla, campañol, ualabi, lobo
Clase1	Fuibat, vampiro
Clase2	
Clase3	pollo, cuervo, paloma, pato, flamenco, pulga, mosquito, gaviota, halcón, abeja, mosca, kiwi, vaquita de sa, alondra, mariposa, avestruz, perico, faisán, ñandú, escorpión, picotijera, pagalo, babosa, gorrión, cisne, termita, tortuga, buitre, avispa, gusano, troglodito
Clase4	almeja, cangrejo, cangrejo de rio, rana, rana macho, langosta, trinon, pulpo, pingüino, pitviper, seawasp, caracol, estrella de mar, sapo, tuatara
Clase5	Carpa, merlango, hipocampo, lenguado
Clase6	róbalo, bagre, cacho, cazon, delfín, arenque, lucio, piraña, puerco, serpiente de mar, pastinaca, atún

Tabla 4. Clasificación realizada

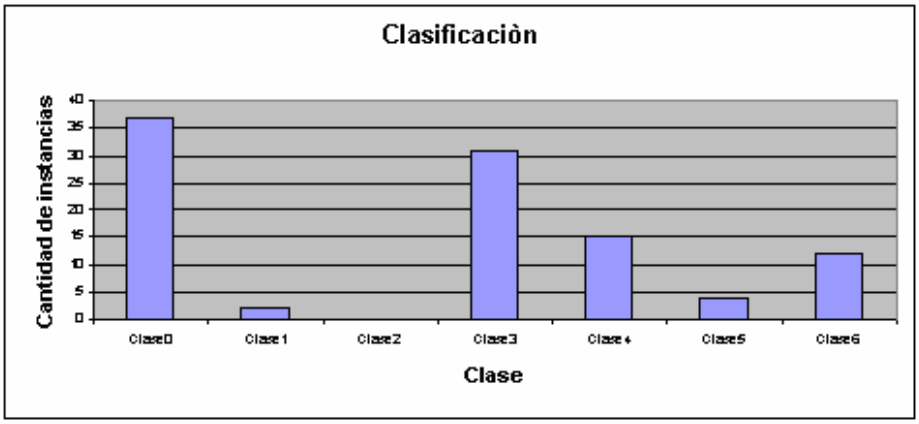


Figura 3. Clasificación de base de datos Zoología

Si generamos las reglas ponderadas, y seleccionamos sólo aquellas reglas con probabilidad de ocurrencia mayor al 70%, obtenemos las reglas detalladas en la tabla 5.

5. CONCLUSIONES

En base a los resultados experimentales obtenidos en la clasificación de diversas poblaciones, en la inferencia de reglas de producción o en la ponderación de la ocurrencia de éstas reglas, podemos concluir que el ambiente desarrollado se comporta en forma similar a otros ambientes existentes en el mercado. El aspecto más importante de este ambiente es que, en contraposición a los ambientes utilizados para la comparación, presenta en su funcionalidad la integración de las tres funciones, aspecto que los ambientes no tienen. Esta característica hace que el ambiente provea una funcionalidad completa para el estudio de las características de una población de individuos, que de acuerdo a las necesidades del usuario, pueden ser las siguientes:

- Clasificar una población y obtener las reglas de producción que dieron como origen a la clasificación.

- Clasificar una población y obtener la probabilidad de ocurrencia de cada regla de producción.
- Si se tiene una población ya clasificada, obtener las reglas de producción que dan como origen a la clasificación..

Reglas de producción

SI domestico=0 y plumas=0 y aerotransportado=0 y respira=0 y huevos=1 y dentado=0
ENTONCES Clase=Clase4
PROBABILIDAD=7859234122635849

SI domestico=0 y plumas=0 y aerotransportado=0 y respira=0 y huevos=1 y dentado=1 y cola=1 y depredador=1
ENTONCES Clase=Clase6
PROBABILIDAD=8207729537464239

SI domestico=0 y plumas=0 y aerotransportado=0 y respira=1 y aletas=0 y patas=cuatro y dentado=1 y huevos=0
ENTONCES Clase=Clase0
PROBABILIDAD=9933109013577036

SI domestico=0 y plumas=0 y aerotransportado=0 y respira=1 y aletas=1 y leche=0
ENTONCES Clase=Clase6
PROBABILIDAD=9613049814245284

SI domestico=0 y plumas=0 y aerotransportado=0 y respira=1 y aletas=1 y leche=1 y dentado=0
ENTONCES Clase=Clase6
PROBABILIDAD=7353075438130839

SI domestico=0 y plumas=0 y aerotransportado=0 y respira=1 y aletas=1 y leche=1 y dentado=1 y huevos=0 y pelo=0
ENTONCES Clase=Clase6
PROBABILIDAD=938963991204266

SI domestico=0 y plumas=0 y aerotransportado=0 y respira=1 y aletas=1 y leche=1 y dentado=1 y huevos=0 y pelo=1
ENTONCES Clase=Clase0
PROBABILIDAD=8284580624335341

SI domestico=0 y plumas=1 y dentado=0 y acuatico=1 y leche=0 y patas=dos y pelo=0 y aletas=0 y venenoso=0 y huevos=0
ENTONCES Clase=Clase3
PROBABILIDAD=9892882462120232

SI domestico=0 y plumas=1 y dentado=0 y acuatico=1 y leche=0 y patas=dos y pelo=0 y aletas=0 y venenoso=0 y huevos=1 y depredador=0
ENTONCES Clase=Clase3
PROBABILIDAD=9995637035200425

SI domestico=0 y plumas=1 y dentado=0 y acuatico=1 y leche=0 y patas=dos y pelo=0 y aletas=0 y venenoso=0 y huevos=1 y depredador=1 cola=0
ENTONCES Clase=Clase3
PROBABILIDAD=9934670737674987

SI domestico=0 y plumas=1 y dentado=0 y acuatico=1 y leche=0 y patas=dos y pelo=0 y aletas=0 y venenoso=0 y huevos=1 y depredador=1 y cola=1 y espina_dorsal=0
ENTONCES Clase=Clase3
PROBABILIDAD=9969610094284359

SI domestico=0 y plumas=1 y dentado=0 y acuatico=1 y leche=0 y patas=dos y pelo=0 y aletas=0 y venenoso=0 y huevos=1 y depredador=1 y cola=1 y espina_dorsal=1 y catsize=0
ENTONCES Clase=Clase3
PROBABILIDAD=99775847706819

SI domestico=0 y plumas=1 y dentado=0 y acuatico=1 y leche=0 y patas=dos y pelo=0 y aletas=0 y venenoso=1
ENTONCES Clase=Clase3
PROBABILIDAD=9942257572349545

SI domestico=0 y plumas=1 y dentado=0 y acuatico=1 y leche=0 y patas=dos y pelo=0 y aletas=1
ENTONCES Clase=Clase3
PROBABILIDAD=9167562568493426

SI domestico=0 y plumas=1 y dentado=0 y acuatico=1 y leche=0 y patas=dos y pelo=1
ENTONCES Clase=Clase3
PROBABILIDAD=7254120295155536

Tabla 5. Reglas con probabilidad de ocurrencia mayor al 70%.

- Si se tiene una población ya clasificada, obtener la probabilidad de ocurrencia de cada regla de producción.

- Se tiene la necesidad de clasificar a una población, pero se necesita que el ambiente provea cual es el atributo por el cual se deba clasificar a la población.
- Permite al usuario la elección de los atributos que se quieren considerar, y sólo utilizar esos atributos en el estudio de las características de la población.
- Permite sólo seleccionar aquellas reglas de producción con una probabilidad de ocurrencia mayor a cierto valor deseado.
- Importancia de poder contar con el Standard XML para la representación de la población, junto con el archivo esquema como entrada también.

6. REFERENCIAS

[1] J. Ross Quinlan. 1986. *Induction of decision trees*. Machine Learning.

[2] J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning.* Machine Learning.

[3] ISoft. 2007. *AC*^{2}. www.alice-soft.com/html/prod_ac2.htm. Vigente al 30/06/2007

[4] SPSS. 2007. *Answer.* /www.spss.com/la/productos/answer-tree/answer.htm. Vigente al 30/06/2007

[5] Salford Systems. 2007. *CART.* www.salford-systems.com/cart.php. Vigente al 30/06/2007

[6] Breiman L, Friedman J, Olshen R y Stone C. 1984. *Classification and regression trees.* Machine Learning.

[7] SPSS. 2007. *Clementine.* www.spss.com/clementine/. Vigente al 30/06/2007

[8] Ian H. Witten y Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques.* 2nd Edition. Morgan Kaufmann. San Francisco.

[9] Kohonen, T. 2001. *Self-Organizing Maps*. 3rd Edition. Springer

APLICACION DE MINERIA DE DATOS PARA LA EXPLORACION Y DETECCION DE PATRONES DELICTIVOS EN ARGENTINA

F. Valenga[1], I. Perversi[2], E. Fernández[2,3], H. Merlino[2,3], D. Rodríguez[2], P. Britos[2,3] y R. García-Martínez[2,3]

[1] Licenciatura en Informática. Universidad de Morón.
[2] Centro de Ingeniería de Software e Ingeniería del Conocimiento. Escuela de Postgrado. ITBA.
[3] Laboratorio de Sistemas Inteligentes. Facultad de Ingeniería. Universidad de Buenos Aires.
Argentina
rgm@itba.edu.ar

Resumen

El presente trabajo describe un Proyecto de Minería de Datos en el ámbito de la información criminal, analizando los homicidios dolosos cometidos en la República Argentina mediante una herramienta de distribución libre.

Palabras claves: Minería de Datos. Inteligencia Criminal.

Abstract

This article describes a Project of Data mining in thee area of the criminal information, analyzing. The fraudulent homicides committed in the Republic Argentina using a tool of free distribution.

Keywords: Data Mining. Criminal Intelligence.

1. Estadistica y Mineria de Datos: Abordajes complementarios

El abordaje metodológico-estadístico utilizado por el análisis e interpretación sobre criminalidad en la Argentina actualmente en uso en la Dirección Nacional de Política Criminal (DNPC) es consistente con la tradición científica en el área [Kumar,1996; Marczyk, *et al*.,2005; Creswell, 2003] y con las metodologías utilizadas a nivel mundial en el área [Chen *et al*., 2004; Zeleznikow, 2005; Coplink; 2007] .
La minería de datos, así como el descubrimiento de conocimientos en los datos, integra desarrollos y concepciones provenientes de la estadística, el aprendizaje automático, la visualización de datos y la teoría de bases de datos. Esta fusión de disciplinas muy diversas ha estado motivada (entre otras) por el significativo incremento del volumen de los datos en todas las esferas de la actividad humana y en este caso particular en la necesidad de disponer de la mayor cantidad de elementos para establecer políticas de inteligencia criminal mas ajustadas con base en los datos disponibles en los diferentes soportes.
Ambos abordajes han mostrado ser complementarios. Mientras que la Estadística plantea hipótesis que deben ser validadas a partir de los datos disponibles, la Minería de Datos descubre patrones en los datos disponibles que mediante la interpretación de expertos del dominio propone patrones de comportamiento social (en nuestro caso) no previstos desde el otro abordaje.
En este contexto la Minería de Datos emerge como el siguiente paso evolutivo en el proceso de análisis de datos criminales.
Para validar la utilidad del uso de minería de datos en la exploración y detección de patrones delictivos y su complementariedad con el abordaje estadístico utilizado en la DNPC se han hecho algunos trabajos exploratorios cuyos resultados se presentan en las siguientes secciones.

2. Estado de la Cuestión

A partir de la crisis de finales de 2001, Argentina se vio afectada por una creciente ola de inseguridad caracterizada por un aumento en los índices delictivos y los niveles de violencia. Esta situación fue más profunda en los principales centros urbanos y llevó a tomar acciones coordinadas a nivel nacional tendientes a prevenir el delito. Una de estas medidas fue la creación del Sistema de Alerta Temprana (SAT) por parte del Ministerio de Justicia y Derechos Humanos. En el plano internacional, los ataques terroristas del 11 de septiembre han aumentado significativamente la preocupación por la seguridad interna en todo el mundo. Agencias de inteligencia como la CIA o el FBI procesan y analizan información activamente en busca de actividad terrorista [Chen *et al*., 2004].
En este contexto, el análisis de los registros criminales es fundamental en la prevención del delito. Entre otras cosas, porque permite el diseño de políticas y planes de prevención efectivos. En Argentina este tipo de análisis se ha realizado históricamente mediante herramientas estadísticas descriptivas o deductivas, considerando fundamentalmente variables y relaciones primarias. Sin embargo, muchas veces la estadística descriptiva clásica no refleja la verdadera interrelación de las variables y por lo tanto, el problema real. Este contexto requiere un tratamiento estadístico más complejo que nos obliga a evolucionar en el análisis de información criminal.
En general, el tamaño de las bases de datos está basado en aspectos como la capacidad y eficiencia de almacenamiento y no en su posterior uso o análisis. Por esta razón, en muchos casos, los registros almacenados son demasiado grandes o complejos como para analizar [Kantardzic, 2002] y superan el alcance de la estadística [Hand, 1997]. La Minería de Datos (*Data Mining*) es un proceso iterativo de búsqueda de información no trivial en grandes volúmenes de datos que busca generar información similar a la que podría generar un experto humano: patrones, asociaciones, cambios, anomalías y estructuras significativas [Britos *et al*, 2005].

En el caso de la inteligencia criminal, la gran cantidad de información y de variables intervinientes justifican el uso de herramientas más potentes que la estadística convencional que permitan determinar relaciones multivariantes subyacentes. La minería de datos aplicada a la inteligencia criminal es un campo bastante nuevo y ha tenido un gran impulso en los últimos años, sobre todo en EEUU [Chen *et al*., 2004].

2.1. Tratamiento de la Información Criminal en el mundo

A continuación se describen algunas de las principales experiencias de aplicación de minería de datos en el análisis de información criminal a nivel mundial:

- *Proyecto COPLINK*

  El Proyecto COPLINK fue creado en el año 1997 en el Laboratorio de Inteligencia Artificial de la Universidad de Arizona, en Tucson, con el objetivo de servir de modelo para ser llevado a nivel nacional. Recientemente se ha desarrollado la versión comercial, denominada *COPLINK Solution Suite* [Coplink, 2007].

  Coplink está compuesto por dos sistemas integrados: Coplink Connect y Coplink Detect. El primero busca compartir información criminal entre distintos departamentos policiales, mediante un fácil acceso y una interfase sencilla, integrando distintas fuentes de información. El segundo esta diseñado para detectar de forma automática distintos tipos de asociaciones entre las bases de datos mediante técnicas de minería de datos. Ambos sistemas presentan una interfase visual amigable [Chen et al., 2004]. Entre otras aplicaciones Coplink provee *Análisis de Redes Criminales* [Chen et al., 2004], la cual consiste en: identificar las redes o bandas criminales, sus líderes o integrantes clave y como se relacionan entre sí. En primer lugar se utiliza la técnica de *concept space* para extraer relaciones de los sumarios policiales y construir una posible red de sospechosos. La fuerza del vínculo entre dos sospechosos se mide en base a la frecuencia de hechos en los que participaron ambos. Luego se utiliza *clustering* jerárquico para partir la red en subgrupos y *block modeling* para identificar patrones de interacción entre los mismos. Finalmente se calcula el baricentro de cada subgrupo para determinar su miembro clave o líder.

- *Proyecto OVER*

  El Proyecto OVER comenzó en el año 2000 en Reino Unido como una iniciativa conjunta de la Policía de West Midlands y el Centro de Sistemas de Adaptación y División de Psicología de la Universidad de Sunderland. El proyecto esta enfocado en los casos de robo a domicilio particulares. Sus principales objetivos son [Zelzenikow, 2005]:

  - identificar los recursos críticos para establecer estrategias de prevención y detección más eficientes;
  - proveer de fundamentos empíricos para el desarrollo de planes interdepartamentales orientados a la reducción del delito;
  - identificar la información relevante a ser recolectada en el lugar del hecho, redundando en mejoras de eficiencia y reducción de tiempo del personal policial;
  - alimentar al sistema tanto con información *hard* (información forense) como *soft* (información sobre la escena del delito);
  - analizar la distribución espacio-temporal de los hechos y confirmar las suposiciones sobre tendencias y patrones.

- *Otras proyectos*

- El Departamento de Policía de Ámsterdam utiliza el software de minería de datos DataDetective [Sentient, 2007] junto con Mapinfo para el análisis de registros criminales. Las principales técnicas empleadas son árboles de decisión y redes neuronales de backpropagation. Han unificado varias bases de datos policiales junto con información externa (clima, variables socioeconómicas y demográficas) en un único data warehouse. Los principales usos son:
  - identificación de las causas del comportamiento criminal (por ejemplo casos de reincidencia);
  - identificación de las causas del delito en un determinado barrio;
  - agrupamiento de delitos parecidos en *clusters* y su descripción, permitiendo un abordaje más efectivo;
  - identificación de delitos parecidos utilizando algoritmos *fuzzy search*, relacionando casos no resueltos con casos resueltos;
  - identificación de zonas de aumento del delito (por ejemplo se ha utilizado para la localización de equipos preventivos en operativos de búsqueda de armas);
  - evaluación de la performance policial.
- El Departamento de Policía de Richmond (Virginia) ha desarrollado una aplicación para el análisis de información criminal que combina minería de datos, mediante el software Clementine [SPSS, 2007], junto a un entorno visual aportado por Information Builders [IB, 2007] y una iterfase desarrollada por RTI Internacional [RTI, 2007]. El principal objetivo es optimizar la alocación de recursos, en base a una modalidad preactiva y no reactiva. Por ejemplo durante año nuevo se identificaron las zonas que habían tenido un aumento en los casos de heridos de con arma de fuego el año anterior y para la noche se reforzaron exclusivamente esas zonas. El resultado obtenido fue una reducción del 49% en los casos de este tipo con un menor requerimiento de personal policial aproximadamente 50 agentes menos) [SPSS, 2007].
- La Policía Estatal de Illinois adquirió en 2005 un software de minería de datos del compañía RiverGlass Inc. [RiverGlass, 2007] con el objetivo de analizar la información criminal en tiempo real. El campo de aplicación es muy grande y va desde la seguridad marítima en los puertos a la detección de casos de fraude financiero.
- El Departamento de Policía de San Francisco desarrolló junto a IBM la aplicación CrimeMaps, en base a la tecnología DB2 de IBM [IBM, 2007]. Este software permite a los oficiales mediante un simple explorador web buscar un determinado tipo de crimen, realizar análisis de clustering y fijar niveles umbrales de alerta temprana para un determinado delito en una determinada zona de acuerdo a una frecuencia histórica.
- El Departamento de Policía de Nueva York inició en julio de 2005 el Real Time Crime Center [NYC, 2007]. Este ambicioso proyecto tiene como objetivo conformar un enorme data warehouse y cruzar información de todo tipo mediante herramientas de inteligencia de negocios (como Repotnet 1.1 y Accurint Pro) de forma de detectar patrones de comportamiento y asociaciones antes desapercibidos.

2.2. La información criminal en la Argentina

La presente sección esta basada en las reuniones mantenidas con personal de la Dirección Nacional de Política Criminal del Ministerio de Justicia y Derechos Humanos de la Nación.
Se entiende por información criminal a toda aquella información resultante a partir de un presunto delito o de sus componentes (victima, victimario, propiedades, vehículos, etc.) que sea relevante

para la toma de decisiones a posteriori. Ya sea en la prevención, detección y esclarecimiento del delito como en la prosecución de delincuentes, la mejora de procesos judiciales y la creación de nuevas leyes. Según esta definición la mayor fuente de información criminal es el Sistema Penal, entendido como el conjunto de instituciones y procedimientos presentes en el proceso que transita un hecho delictuoso desde que es registrado por el Estado.
Se puede subdividir al Sistema Penal según las distintas instancias en: Sistema Policial, Sistema Judicial y Sistema Penitenciario. Como muestra la Figura 1, una gran cantidad de hechos ingresa por el Sistema Policial, atraviesa el cuello de botella del Sistema Judicial y egresa a través del Sistema Penitenciario.

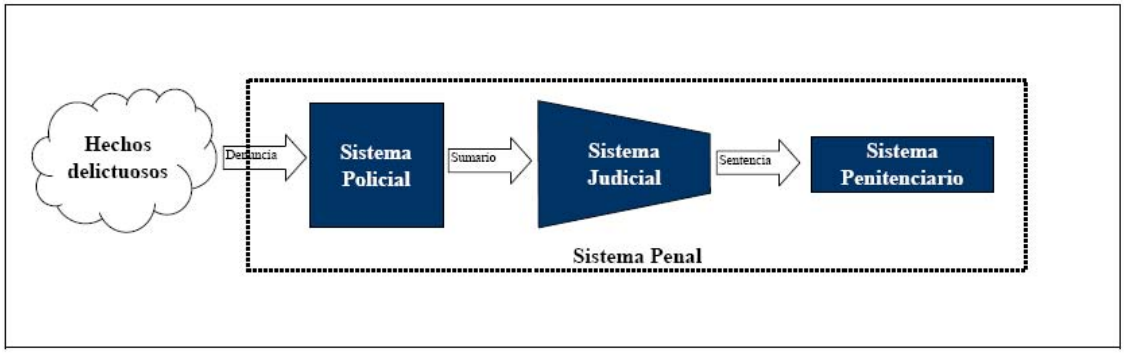


Figura 1- Esquema del Sistema Penal

En Argentina debido al sistema de gobierno federal adoptado, este esquema se replica en cada una de las provincias del territorio nacional.
Este sistema de gobierno tiene fuertes implicancias en la consolidación de la información a nivel nacional. No sólo por la falta de homogeneidad entre las distintas provincias, sino fundamentalmente porque cada provincia tiene autonomía sobre la información generada bajo su jurisdicción y el Estado Nacional no tiene injerencia sobre la misma.
Las funciones de consolidación de información criminal a nivel nacional y confección de la estadística general fueron delegadas, a través de la Ley Nacional 25.266, a la Dirección Nacional de Política Criminal (DNPC) del Ministerio de Justicia y Derechos Humanos de la Nación (MJDHN). Esta ley establece a la DNPC como única fuente oficial de información criminal a nivel nacional. Cabe mencionar que esta capacidad de la DNCP no va en desmedro de que las distintas instituciones del Sistema Penal posean su propia información, e incluso sectores de estadística, para analizar su propia gestión. De hecho, la gran mayoría de las policías nacionales poseen una división de estadística.
Las primeras estadísticas policiales en Argentina comenzaron a ser elaboradas por la Policía de la Capital Federal en 1887 [Blackwelder Y Jonson, 1984]; [Rubial, 1993]; [Sozzo, 2000]. A partir de 1971 toda la información proveniente del Sistema Policial pasó a ser consolidada a nivel nacional por el Registro Nacional de Reincidencia y Estadísticas Criminales (RNREC). Los siguientes 30 años se caracterizaron por estadísticas incompletas, de poca calidad y sin un análisis posterior.
A partir de 1999 se intentó revertir esta situación mediante la creación del Sistema Nacional de Información Criminal (SNIC) y el Sistema de Alerta Temprana (SAT), y la transferencia de las funciones de consolidación y análisis de información criminal a la Dirección Nacional de Política Criminal (DNPC). En julio de 2000 se formalizó esta transferencia mediante la Ley Nacional 25.266 anteriormente mencionada.
El Sistema de Alerta Temprana (SAT) se nutre de cuatro planillas complementarias a la del SNIC que relevan información detallada sobre cuatro aspectos en particular:

- homicidios dolosos;
- homicidios culposos en accidentes de tránsito;
- suicidios;

- delitos contra la propiedad.

En los primeros tres casos se releva información puntual de cada hecho, mientras que el último consta de sumarizaciones parciales según distintas variables. La información relevada en cada caso es la siguiente.

2.3. Introducción a las técnicas de Minería de Datos

La Minería de Datos es un proceso iterativo de búsqueda de información no trivial en grandes volúmenes de datos [Kantardzic, 2002] que busca generar información similar a la que podría generar un experto humano: patrones, asociaciones, cambios, anomalías y estructuras significativas [Britos et al, 2005].
Las principales técnicas de minería de datos se suelen clasificar según su tarea de descubrimiento, en tal sentido a continuación se describen las clasificaciones consideradas más relevantes para el actual proyecto:

- Agrupación o clustering.
- Clasificación.

A continuación se realiza una breve descripción de cada una de estas clasificaciones:

- *Agrupación de Datos o Clustering:*

  La agrupación o el clustering consiste en agrupar un conjunto de datos, sin tener clases predefinidas, basándose en la similitud de los valores de los atributos de los distintos datos. Esta agrupación, a diferencia de la clasificación, se realiza de forma no supervisada, ya que no se conoce de antemano las clases del conjunto de datos de entrenamiento. El clustering identifica clusters, o regiones densamente pobladas, de acuerdo a alguna medida de distancia, en un gran conjunto de datos multidimensional [Chen y Han, 1996]. El clustering se basa en maximizar la similitud de las instancias en cada cluster y minimizar la similitud entre clusters [Han y Kamber, 2001].

  K-Means [Britos et al., 2005] en un método particional de clustering donde se construye una partición de una base de datos D de n objetos en un conjunto de k grupos, buscando optimizar el criterio de particionamiento elegido. En K-Means cada grupo está representado por su centro. K-Means intenta formar k grupos, con k predeterminado antes del inicio del proceso. Asume que los atributos de los objetos forman un vector espacial. El objetivo que se intenta alcanzar es minimizar la varianza total intra-grupo o la función de error cuadrático.

- *Clasificación de Datos:*

  La clasificación se utiliza para clasificar un conjunto de datos basado en los valores de sus atributos. Por ejemplo, se podría clasificar a distintas personas para el otorgamiento de un préstamo en riesgo bajo, medio y alto, teniendo en cuenta información histórica de las mismas.

  La clasificación encuentra las propiedades comunes entre un conjunto de objetos y los clasifica en diferentes clases, de acuerdo a un modelo de clasificación. Para construir este modelo, se utiliza un conjunto de entrenamiento, en el que cada instancia consiste en un conjunto de atributos y el valor de la clase a la cual pertenece [Chen y Han, 1996].

  Los algoritmos mayormente utilizados para las tareas de clasificación son los algoritmos de inducción. En la actualidad existen numerosos enfoques de algoritmos de inducción y variedad en cada enfoque, el presente trabajo hará hincapié en aquellos orientados a generar árboles de decisión.

  ID3 es un sistema típico de construcción de árboles de decisión, el cual adopta una estrategia de arriba hacia abajo e inspecciona solo una parte del espacio de búsqueda. ID-3 garantiza que será encontrado un árbol simple, pero no necesariamente el más simple. ID-3 utiliza la teoría de la información para minimizar la cantidad de pruebas para clasificar un objeto. Una heurística selecciona el atributo que provee la mayor ganancia de la información. Una extensión a ID3, C4.5 [Weka, 2007] extiende el dominio de clasificación de atributos categóricos a numéricos. J48 es una implementación mejorada del algoritmo de árboles de decisión C4.5. El algoritmo J48 funciona bien con atributos nominales y numéricos. Un paso importante en la construcción del árbol de decisión es la poda, la cual elimina las ramas no necesarias, resultando en una clasificación más rápida y una mejora en la precisión de la clasificación de datos [Han y Kamber, 2001].

Existen en la actualidad varias herramientas de libre distribución que permiten aplicar las técnicas antes mencionadas, entre ellas se encuentra *Weka* [Weka, 2007]. La cuan fue desarrollada originalmente en la universidad de Waikato y hoy día es accesible fácilmente desde Internet.

2.4. Antecedentes vinculados al tratamiento de la información criminal en la Argentina

En Argentina no se conoce ninguna experiencia de aplicación de minería de datos a información criminal. Sin embargo hay dos proyectos relacionados que merecen ser mencionados.

- *El Proyecto SURC*

  A comienzos de 2004 el entonces Ministerio de Justicia, Seguridad y Derechos Humanos lanzó el proyecto del Sistema Unificado de Registros Criminales (SURC). El objetivo era interconectar y articular las instituciones del Sistema Policial y el Sistema Judicial mediante una red en la cual todos tuvieran acceso a un banco de datos común, alimentado en tiempo real y del cual se pudieran realizar consultas *online*. Este banco de datos contemplaba información diversa [SSI-MI, 2004]:

  - Registro de hechos: características generales del hecho denunciado (lugar, día, hora, delito denunciado y comisaría interviniente).
  - Registro de denunciantes: identidad y características de la víctima o denunciante.
  - Registro de autores identificados: identidad, características, historial criminal e imágenes de los autores.

- Registro de autores no identificados: descripción de los NN (contextura física, edad aproximada, estatura, color de pelo, señas particulares, frases frecuentes, etc.).
- Registro de elementos robados: información útil para la identificación de los objetos robados.
- Registro de autos robados: marca, modelo, color, número de patente, número de motor, características particulares, etc.
- Registro de armas secuestradas: características de las armas secuestradas, vinculando esta base con otros sistemas como el Ibis.
- Registro de evidencias: descripción de huellas y pistas relevadas en la escena del crimen.
- Mapa del delito: presentación de los hechos en forma gráfica y geo-referenciada mediante GIS.

Este proyecto de gran alcance contemplaba una implementación progresiva, comenzando por la Ciudad Autónoma de Buenos Aires y avanzando hacia las provincias. Sin embargo tras la salida del entonces Ministro de Justicia, Seguridad y Derechos Humanos, Gustavo Béliz, y gran parte de su equipo de trabajo, en julio de 2004, el proyecto quedó congelado. Tiempo más tarde se trasladaron las funciones de seguridad de la esfera del Ministerio de Justicia al Ministerio del Interior, y consigo el proyecto SURC. Actualmente el proyecto permanece vigente, con radicación en la Secretaría de Inteligencia Criminal del Ministerio del Interior de la Nación, pero relegado y con un cambio de enfoque respecto al original.

- *El Mapa del Delito de la Ciudad Autónoma de Buenos Aires*

  El Ministerio Público Fiscal de la Nación (MPFN) es una de las pocas instituciones judiciales de Argentina que posee un sistema de información digitalizada. Cuenta con una base de datos de los hechos delictivos de autoría desconocida (NN) registrados en Capital Federal. Esta base contiene información referida al hecho, como ser: tipo de delito, fecha, lugar y cantidad de víctimas.
  Asimismo el Centro de Información Metropolitana (CIM), radicado en la Facultad de Arquitectura, Diseño y Urbanismo (FADU) de la Universidad de Buenos Aires (UBA), posee el Sistema de Información Territorial del Área Metropolitana de Buenos Aires (SAT/AMBA). Este sistema consiste en la base cartográfica digital de todo el AMBA para ser utilizada bajo GISs (*Geographical Information Systems*). No sólo posee los elementos tradicionales (calles, avenidas, vías del ferrocarril, plazas, etc.) sino también la visualización de las demarcaciones zonales (barrios, centros de gestión y participación, comisarías, etc.) y gran parte del equipamiento urbano (escuelas, clubes, bancos, etc.).
  En 2002 ambas instituciones firmaron un "Convenio de Asistencia, complementación y Cooperación" con el objetivo de que el CIM elaborase el Mapa del Delito de la Ciudad Autónoma de Buenos Aires con la información suministrada por el MPFN.
  Si bien la existencia de este mapa no es muy conocida y pese a que cuenta con ciertas limitaciones (únicamente hechos ocurridos en Capital Federal con autoría desconocida), su aporte en el análisis de la situación criminal es muy valioso[Behar, Lucilli, 2003].

3. Definición del problema

El Sistema de Alerta Temprana (SAT) creado por la Dirección Nacional de Política Criminal (DNPC) del Ministerio de Justicia y Derechos Humanos (MJDH) consiste en un relevamiento a nivel nacional de determinados tipos de delitos. En la actualidad esta información es analizada mediante un análisis estadístico básico, sin hacer un aprovechamiento exhaustivo de la información mediante el uso de técnicas o herramientas de Minería de Datos.

4. Solución propuesta

Para aumentar el poder de la información criminal existente, se propone llevar a delante un proyecto de Minería de Datos mediante la herramienta Weka 3.5.3. tomando como base los datos reportados por el sistema SAT.

A continuación se describe la estrategia de aplicación de algoritmos:

1. En primer lugar se prevee aplicar el algoritmo Simple K-means para clusterizar. Estos resultados serán analizados y convalidados con los usuarios, utilizando para ello los informes emitidos por Weka.
2. Una vez estabilizados los cluster se utilizarán algoritmos de Inducción para explicar el comportamiento de los mismos de una forma mas descriptiva. Para ello se utilizará el algoritmo J48.

5. Demostración de la Solución

5.1. Descripción del Dataset

Se analizaron 1810 registros de la base de datos “Homicidios Dolosos” correspondientes a la totalidad de hechos registrados durante 2005, provenientes del SAT. Cuyos atributos se describen a continuación en la tabla 1:

Provincia	Departamento	Día del mes	Mes	Día de la semana
Hora	Lugar	Arma	Otro delito	

Tabla 1- Atributos del Dataset

5.2. Resultados del proceso de Agrupamiento

5.2.1. Centroides

A continuación, en la tabla 2, se describen los centroides obtenidos:

		Atributos categóricos (modas)				Atributos continuos (medias)			
	Cant. (%)	Provincia	Lugar	Arma	Otro Delito	Hora	Día Semana	Día Mes	Mes
Cluster 0	22%	BsAs	Vía Pública	de Fuego	Robo	19	Sábado	16	7
Cluster 1	43%	BsAs	Vía Pública	de Fuego	No Hubo	17	Sábado	15	7
Cluster 2	35%	BsAs	Domicilio Particular	Blanca	No Hubo	21	Sábado	15	7
General	100%	BsAs	Vía Pública	de Fuego	No Hubo	19	Sábado	15	7

Tabla 2- Centroides

5.2.2. Gráficos de Barras

La distribución de los clusters entre las variables de los distintos atributos permite comprender el nivel de significancia de los mismos (ver figura 2). En este caso, si los clusters fueran irrelevantes, esperaríamos encontrar una proporción aproximada de 43% rojo (cluster 1); 22% azul (cluster 0) y 35% turquesa (cluster 2) en cada variable de cada atributo. Si bien en algunos atributos esta

proporción se cumple (*día mes y provincia*) en otros existen interacciones significativas (por ejemplo cluster 2 con *arma blanca* y *domicilio particular*):

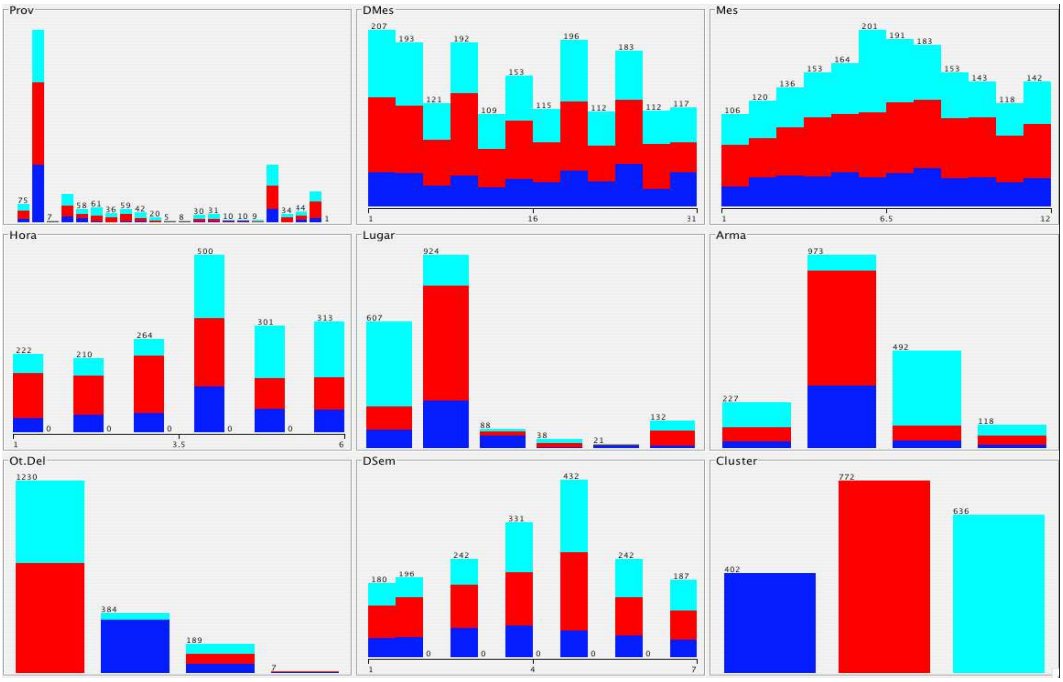


Figura 2- Distribución de Clusters

5.2.3. Gráficos de dispersión

A continuación se describen los cluster en base a dos de los atributos más representativos:

Distribución de los clusters según el atributo lugar (ver figura3) : Mientras el cluster 2 esta muy concentrado en domicilio particular y el cluster 1 en vía pública, el cluster 0 se encuentra distribuido más homogéneamente [Figura 3]. Si bien este último presenta la mayoría de registros en domicilio particular, tiene una alta proporción de homicidios en comercios respecto a los otros clusters.

Distribución de los clusters según el atributo arma (ver figura 4) : El cluster 1 y el cluster 0 presentan una distribución similar, con una alta concentración en arma de fuego, seguida por arma blanca y prácticamente muy pocos casos sin arma [Figura 4]. En contraposición el cluster 2 presenta muy pocos casos con arma de fuego (una proporción muy baja respecto a la proporción global) y muchos casos sin arma (una proporción muy alta respecto a la proporción global).

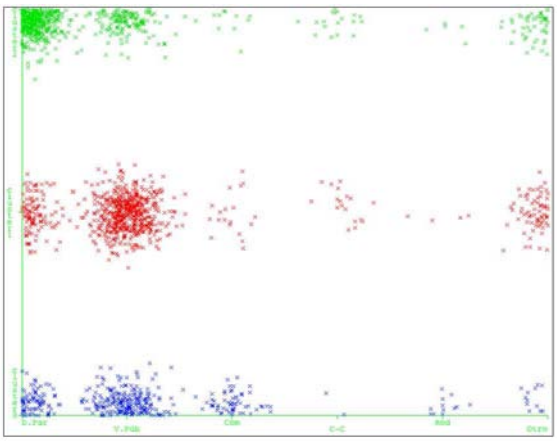


Figura 3- Distribución según atributo lugar

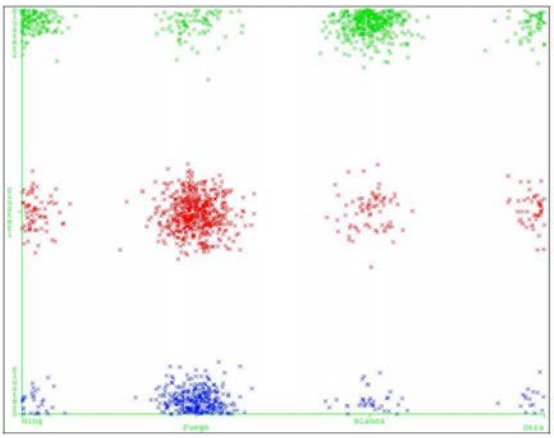


Figura 4- Distribución según atributo Arma

5.2.4. Análisis preliminar de los cluster

En base a la información que surge de este análisis podemos dar una primera interpretación a los clusters:

- *Cluster 0 (22%): esta caracterizado por homicidios mayoritariamente en ocasión de robo y con arma de fuego. En principio diremos que se trata de "homicidios en ocasión de robo".*
- *Cluster 1 (43%): es el que más registros agrupa y el más parecido a la media global. Está caracterizado por homicidios mayoritariamente en la vía pública con arma de fuego y sin la existencia de otro delito. Se podrían interpretar como "homicidios en ocasión de riña o ajuste de cuentas".*
- *Cluster 2 (35%): es el más particular de los clusters, ya que la mayoría de sus registros presentan casos de homicidios sin arma de fuego y en domicilio particular. Los denominaremos "homicidios en ocasión de emoción violenta".*

5.3. Árbol de clasificación

Como el árbol clasificatorio obtenido es muy grande, a continuación, en la tabla 3, se describe el 9% de las reglas de clasificación extraídas del árbol, las cuales clasifican el 66% de las instancias (1200). Estas reglas son las siguientes:

Regla 1 SI otro delito = no hubo Y arma = fuego Y lugar = V.Púb. ENTONCES Cluster 1 (362)	**Regla 2** SI otro delito = robo Y arma = fuego ENTONCES Cluster 0 (272)	**Regla 3** SI otro delito = no hubo Y arma = blanca Y lugar = D.Part. ENTONCES Cluster 2 (134)
Regla 4 SI otro delito = no hubo Y arma = blanca Y lugar = V.Pub. Y DSem = Sa-Ma Y hora = 19-8 ENTONCES Cluster 2 (109)	**Regla 5** SI otro delito = no hubo Y arma = ninguna Y lugar = D.Part. ENTONCES Cluster 2 (87)	**Regla 6** SI otro delito = no hubo Y arma = fuego Y lugar = D.Part. Y hora = 8-19 ENTONCES Cluster 1 (85/3)
Regla 7 SI otro delito = no hubo Y arma = blanca Y lugar = V.Pub. Y hora = 8-16 ENTONCES Cluster 1 (55)	**Regla 8** SI otro delito = no hubo Y arma = ninguna Y lugar = V.Pub. ENTONCES Cluster 1 (48)	**Regla 9** SI otro delito = no hubo Y arma = fuego Y lugar D.Part. Y hora = 20-8 ENTONCES Cluster 1 (48)

Tabla 3- Reglas de Clasificación

Estas reglas fueron consultadas con los especialistas y permitieron confirmar la interpretación hecha anteriormente. Al respecto, los especialistas comentaron que hasta el momento ellos solían clasificar a los homicidios en dos grupos, según el vínculo existente entre la víctima y el agresor:

- *los casos de robo, en los que víctima y agresor no se conocen;*
- *el resto de los casos, denominados "homicidios en conflictos interpersonales".*

La conclusión arribada junto con los especialistas es que se trata de **dos tipos de conflictos interpersonales distintos**, uno más bien **familiar** (dentro del domicilio particular) e **impulsivo** (sin

arma) y otro más bien **vecinal** o de **ajuste de cuentas** (vía pública) y con cierto nivel de **premeditación o pre-intencionalidad** (arma de fuego).
En el medio de estos dos grupos extremos están los casos de armas blancas, difíciles de asignar a priori a una u otra modalidad

6. Conclusiones

El presente trabajo ha demostrado no sólo que es factible aplicar minería de datos a la información criminal en Argentina, sino también su alto valor agregado para el análisis y la generación de nuevo conocimiento.
La experiencia realizada en conjunto con la Dirección Nacional de Política Criminal (DNPC) del Ministerio de Justicia y Derechos Humanos de la Nación (MJDHN), basa la factibilidad en los siguientes puntos:

- *existe gran cantidad de información que actualmente no esta siendo aprovechada en toda su dimensión;*
- *existe un software de minería de datos de distribución libre y gratuita, fácil de usar y que contiene las herramientas necesarias para el análisis;*
- *este software de minería de datos puede ser utilizado por una persona ajena al ámbito informático con una capacitación básica.*

Los resultados experimentales obtenidos han sido validados por los especialistas de la DNPC. Estos resultados han permitido tanto confirmar conceptos preexistentes (pero con una justificación sustentada en los datos), como generar nuevas piezas de conocimiento. Al respecto se han identificado tres patrones distintos de homicidios dolosos en base a los hechos ocurridos en Argentina durante 2005.

7. Futuras líneas de investigación

En primer lugar se propone aumentar el alcance de la información de la DNPC a ser analizada con este tipo de técnicas. Esto implica tanto una expansión transversal, haciendo uso de otras bases de datos como la de “homicidios culposos en accidentes de tránsito”; como longitudinal, analizando la información histórica existente para detectar patrones de evolución temporal en cuanto a las modalidades delictivas.
En segundo lugar se sugiere el diseño de procedimientos estándar de minería de datos con *Weka* para ser implementados en la DNPC. Esta batería de procedimientos les permitiría a los analistas de la DNPC extraer e identificar patrones y asociaciones en forma automatizada y estandarizada.
En tercer lugar se propone proceder al análisis de la información geográfica relevada por la DNPC (que hoy no es aprovechada) mediante GISs (*Geographical Information Systems*). Este tipo de análisis permitiría detectar, por ejemplo, zonas de alta densidad de homicidios en accidentes de tránsito.
Finalmente se propone expandir el uso de estas técnicas a las fuerzas de seguridad, en donde estas aplicaciones han encontrado su mayor aplicación a nivel mundial.

8. Referencias

Behar, A. M., P. Lucilli, 2003. *Mapa del delito de la Ciudad Autónoma de Buenos Aires*. Terceras Jornadas de Jóvenes Investigadores, Instituto Gino Germani.

Blackwelder, J.K., L.L. Jonson, 1984. *Estadística Criminal y Acción Policial en Buenos Aires, 1887-1914*. Desarrollo Económico, 93, Vol. 24, 1984, pp. 109-122.

Britos, P., Hossian, A., García-Martinez, R. y Sierra, E. 2005. Minería de Datos Basada en Sistemas Inteligentes. Editorial Nueva Librería.. ISBN 987-1104-30-8 .

Chen, H., W. Chung, J. Xu, G. Wang, Y. Qin, M. Chau, 2004. *Crime Data Mining: A General Framework and Some Examples*. IEEE Computer Society, vol. 37, no. 4, pp. 50-56.

Chen, M., Han, J., *Data mining: An overview from database perspective*. IEEE Transactions on Knowledge and Data Eng., 1996.

Coplink, 2007. COPLINK Solution Suite. www.coplink.com. Acceso mayo 2007.

Creswell, J. 2003. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications.

Han, J., Kamber, M.; *Data mining: Concepts and techniques*. Morgan Kauffmann Publishers, 2001.

Hand, D. J., 1997. *Data Mining: Statistics and More?.* The American Statistician.

IB, 2007. Information Builders. http://www.informationbuilders.com. Acceso mayo 2007.

IBM, 2007. Internacional Business Machines. http://www03.ibm.com/industries/goverment/doc/content/news/pressrelease/1019264109.html Acceso mayo 2007.

Kantardzic, M. 2002. *Data Mining: Concepts, models, methods and algorithms.* Wiley-IEEE Press. ISBN 0-471-22852-4.

Kumar, R.; 1996. *Research Methodology: A Step-by-Step Guide for Beginners*. Addison Wesley.

Marczyk, G., DeMatteo, D., Festinger, D.; 2005. *Essentials of Research Design and Methodology* (Essentials of Behavioral Science). John Wiley & Sons.

NYC, 2007. *New York Police Department Real Time Crime Center*. http://www.nyc.gov/html/nypd/html/dcpi/RTCCRevisedFINALWEB_files/frame.htmAcceso mayo 2007.

Quinlan, J. R., *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

RiverGlass, 2007. RiverGlass Inc. http://www.riverglassinc.com/solutions/intelligence.php Acceso mayo 2007.

RTI, 2007. Research Triangle Institute. http://www.rti.org . Acceso mayo 2007.

Rubial B.C., 1993. *Ideología del Control Social, 1880-1920*. Centro Editor de América Latina, Buenos Aires, Argentina.

Sentient, 2007. Sentient Information Systems. http://www.sentient.nl. Acceso mayo 2007.

Sozzo, M., 2000. *Pintando a Través de Números: Fuentes Estadísticas de Conocimiento y Gobierno Democrático de la Cuestión Criminal en Argentina*. http://www.ilsed.org/index.php?option=com_docman&task=doc_view&gid=159&itemid=44 Acceso mayo 2007.

SPSS, 2007. SPSS Inc. URL:http://www.spss.com/success/pdf/CS%20%20Richmond % 20 PD%20LR.pdf. Acceso mayo 2007.

SSI-MI, 2004. *Presentación Institucional Proyecto SURC*. Secretaría de Seguridad Interior, Ministerio del Interior de la República Argentina.

Weka, 2007. *Data Mining Software in Java*; http://www.cs.waikato.ac.nz/ml/weka/ Acceso mayo 2007.

Zeleznikow, J., 2005. *Using Data Mining to Detect Criminal Networks*. www.aic.gov.au/conferences/occasional/2005-04.zeleznikow.html. Acceso mayo 2007.

Aplicaciones de ingeniería de software centradas en el contexto y su uso en autopistas inteligentes.

Analía Gabriela Bigatti, Hugo Orlando Messina, Jorge Salvador Ierache

Facultad de Informática Ciencias de la Comunicación y Técnicas Especiales,
Universidad de Morón
Morón B1708JPD, Buenos Aires, Argentina
analiabigatti@unimoron.edu.ar, hmessina@unimoron.edu.ar, jierache@ unimoron.edu.ar

Abstract

The context-aware systems open an enormous fan of possibilities in the field of application of the traditional engineering of systems. With these systems the user becomes the center of attention of the applications, since these modify their possibilities of election according to the surroundings that cross the user at this moment. The intelligent higways are not the exception. With thousand of car in different situations and into a changing enviroment, it is not enaugh to improve the system of message panels and cabins of toll to avoid delays and accidenes. The present article shows the gap between systems that use context-aware computing of which no, in the scene of the intelligent highways.

Resúmen

Los sistemas centrados en contexto abren un enorme abanico de posibilidades en el campo de aplicación de la ingeniería de sistemas tradicional. Con estos sistemas el usuario pasa a ser el foco de atención de las aplicaciones, ya que estos modifican sus posibilidades de elección de acuerdo al entorno que atraviesa el usuario en este momento. Las autopistas inteligentes no son la excepción. Con miles de coches en situaciones diferentes y un entorno en extremo cambiante hacen que no alcance solo con sofisticar los sistemas de carteles y las cabinas de peaje para evitar demoras y accidenes. El presente artículo muestra la brecha que hay entre sistemas que utilizan computación context-aware frente a los que no la utilizan, en el escenario de las autopistas inteligentes.

Palabras claves: Context-Aware, Pervasive Computing, Ubiquitous Computing, Smart Highway

1 INTRODUCCION

Hoy en día hay un desarrollo creciente en el área de sistemas centrados en contexto (aquellos sistemas capaces de censar, interpretar y responder de acuerdo con el entorno en que se encuentra el usuario, y que hacen uso del contexto para proveer información o servicios relevantes al mismo, en

relación a la tarea que éste se encuentre desarrollando). Como ejemplo de aplicación, nos centraremos en la posibilidad de brindar información contextual a los conductores de vehículos sobre las autopistas, a fin de brindar mayor seguridad frente a situaciones como la que reflejaron los hechos pasados en la tarde del 14 de abril de 2007, cuando un desprevenido conductor estacionó su camioneta de pasajeros en medio de un denso banco de niebla, siendo embestido por un camión y subsecuentemente por otros vehículos, provocando un accidente fatal múltiple en el autopista Ezeiza-Cañuelas sobre el Km. 48.

2. PROBLEMA

El crecimiento de accidentes agravados por muerte, los pocos recursos económicos del país, el avance tecnológico, la posibilidad de lograr que el conducir sea algo placentero y la posibilidad de facilitar el flujo de información vital para los conductores, fueron los ejes que motivaron el desarrollo del presente artículo. Se destacan los principales factores que intervienen en los accidentes entre estos se encuentran los conductores estos influyen principalmente con comportamiento, la falta de atención al camino y al tránsito circundante, las actitudes de no ceder el paso, el incumplimiento de las reglas de tránsito y la inexperiencia y negligencia.

En relación a los vehículos estos influyen principalmente por sus condiciones mecánicas, frenos defectuosos y cubiertas lisas. El camino influye principalmente por las condiciones y calidad del mismo, el tipo de pavimento, cunetas, lomos de burro.

El sistema de control de tránsito, influye en el diseño de las autopistas, su señalización, los obstáculos, los animales sueltos. El medio ambiente influye en relación a la visibilidad, al estado del pavimento mojado, congelado, entre otros.

3. SITUACIÓN ACTUAL

Hoy en día los automovilistas soportan largas filas en las estaciones de peajes, las autopistas se corrompen en horas pico, (entre las 8:00 AM y las 18:00 PM), sumado a este problema las imprudencias automovilísticas han crecido notablemente respecto de años anteriores provocando un importante incremento en la suma de victimas inocentes. Algunas autopistas cuentan con sistemas TAG o Televía, este sistema de cobro automático electrónico agiliza el paso de los vehículos por las estaciones de peajes, aunque presentan inconvenientes en la respuesta de la apertura de barreras y no todas las autopistas se encuentran equipadas con este dispositivo. En los últimos años se han implementado carteles de información visual que al estar en contacto con una central de datos se actualizan a cada minuto brindando el estado de tránsito, en cuanto a demoras, embotellamientos o accidentes, posibles desvíos, rutas alternativas, las velocidades a las que se debe transitar (según el flujo de autos) y servicios meteorológicos.

4. DESCRIPCIÓN SOBRE CONTEXT-AWARE (CENTRADO EN EL CONTEXTO)

4.1 Consideraciones acerca de context-awareness.

A lo largo del tiempo se han formulado diferentes definiciones. La primera definición surge en 1993, (Schilit [22]) define al contexto como "*la localización, identidad de las personas y objetos cercanos y los cambios que se produzcan en estos objetos*", en forma similar, Brown [5] define al contexto como "*localización, identidad de las personas que rodean al usuario, hora del día, estación del año, temperatura, etc*.". Ryan [20] lo presenta como *"la localización del usuario, el entorno físico, la identidad y la hora".* Siguiendo esta línea, en cuanto a la percepción de los elementos que deben ser considerados parte del contexto, Dey [10], en 1998, brinda una definición más completa, donde ve al contexto como *"el estado emocional del usuario, el foco de atención, la localización y orientación, fecha y hora, y las personas que componen el entorno del usuario".*
Todas estas definiciones enumeran los "elementos", que para el autor, tendrían que formar parte del contexto, ahora, ¿qué ocurre si se agrega un nuevo elemento?, ¿como saber si forma o no parte del contexto? Un ejemplo de esto podría ser, *"servicios disponibles".* Al Agregar un nuevo elemento, estas definiciones se vuelven insuficientes y dejan "una zona gris", por tal motivo es necesario encontrar una definición más abstracta y amplia. Brown [4] [6], expreso al contexto como *"los elementos del entorno del usuario de los que la computadora del usuario es consciente"*. Ward [23], lo define sencillamente como "*el estado del entorno de la aplicación"*. Pascoe [16], [17], lo expresa como "*el conjunto de estados físicos y conceptuales de interés para una entidad particular*". La definición de Franklin [12] es sumamente abstracta, caracterizo al contexto como "*la situación del usuario*". Dey [9], aporta una definición desde el punto de vista del desarrollo de software, caracterizando al contexto como: "*cualquier información que puede ser usada para caracterizar la situación de una entidad. Una entidad puede ser una persona, un lugar o un objeto que es considerado relevante para la interacción entre el usuario y la aplicación; incluyendo al usuario y la aplicación mismos".*

4.2 Entorno de aplicaciónes context-awareness

Todos los seres vivos interaccionan constantemente con su entorno. Con múltiples grados de complejidad, los organismos son capaces de percibir los cambios que se producen en el entorno que los rodea y de reacomodar su comportamiento o su metabolismo para adaptarse a dichos cambios. Un cambio que permita a un ser vivo funcionar eficientemente se llama adaptación. El cambio adaptativo significa una ventaja para vivir en un entorno concreto. B. Schilit y M. Theimer [22] y más tarde (1996), M. Brown [4], todos utilizaron el concepto de "*adaptación*" y coincidiendo en una primera definición que expresaba lo siguiente: "*una aplicación de software es sensible al contexto si puede adaptarse, percibir o responder ante los cambios en el entorno*".

Con el tiempo a las aplicaciones context-awareness se les atribuyeron términos como: "*reactivas*" [20], "*obedientes*" [8], "*situadas*" [11], "*sensible al contexto*" [13] y "*dirigidas por el ambiente*" [18]. Kortuem [15], tomo algunos de estos conceptos y definió este tipo de aplicaciones como "*aquellas que pueden variar o adaptar dinámicamente su comportamiento en base al ambiente*". Una aplicación que varia dinámicamente su comportamiento evidentemente está dirigida o/y es sensible al contexto para poder adaptarse a él. Más adelante, entre 1997-1998, Hull [13] y Pascoe [17] definieron a las aplicaciones context-awareness por *"la habilidad que poseen los dispositivos computacionales de detectar, censar, interpretar y responder a los aspectos del ambiente del usuario".* Dey [14], en 1998, continua este camino y define la noción de las aplicaciones context-

awareness como "*el uso del contexto para automatizar un sistema de software, modificar su interface y proveer la máxima flexibilidad en términos de servicios*". Salber [21] añade "*to be the ability to provide maximum flexibility of a computational service based on real-time sensing of context*", estas definiciones se centran en el conocimiento de determinados aspectos del contexto para brindar flexibilidad de servicios al usuario. Por otro lado, Ryan [19], define a las aplicaciones context-awareness como "*aplicaciones que supervisan las entradas provenientes de los dispositivos de censado ambientales y permiten a los usuarios seleccionar un determinado contexto físico o lógico de acuerdo a sus intereses o sus actividades".* Esta definición es más restrictiva que las anteriores por que identifica el método por el cual las aplicaciones actúan sobre el contexto y sobre los intereses del usuario. Siguiendo esta línea, Brown [7] expresa, "*aplicaciones que automáticamente proporcionan información y/o reaccionan de acuerdo contexto actual del usuario que es detectado por los sensores".* Esta definición se centra en el contexto del usuario, es decir, toda la información relevante que influya sobre las intensiones del usuario; esta información provocan una reacción por parte de la aplicación que se ajusta a ese contexto. Este tipo de aplicaciones buscan ser proactivas, anticipándose a las acciones que puede llegar a realizar el usuario, un por ejemplo de esto podría ser, modificar la configuración de un "*layout*" grafico, para esto la aplicación deberá, por ejemplo, tener en cuenta aspectos contextuales como, la luminosidad del ambiente, el horario, la cantidad de batería restante (si es un dispositivo móvil), la actividad que está realizando el usuario en ese momento, etc.

Finalmente, Dey [9] [2] expresa una aplicación context-awareness como: *"aquella que utiliza al contexto para proveer información relevante y servicios al usuario, donde la relevancia depende de la tarea que está llevando a cabo el usuario".*Contexto es cualquier información que puede ser usada para caracterizar la situación de una entidad. Donde una entidad es una persona, lugar u objeto que es relevante para la interacción entre un usuario y una aplicación, incluyendo al usuario y la aplicación. Las nociones de context-aware provienen de las ideas acerca de la computación ubicua y computación pervasiva, cuya finalidad es lograr que el entorno cotidano y los dispositivos de computación estén relacionados. Según su ideador Mark Weiser *las tecnologías más profundas son las que desaparecen. Se tejen en la tela de la vida diaria hasta que son indistinguibles de ella*. [24] Para la adquisición de datos se utilizan sensores que toman del contexto los eventos generados determinadas situaciones como se muestra en la figura 1.

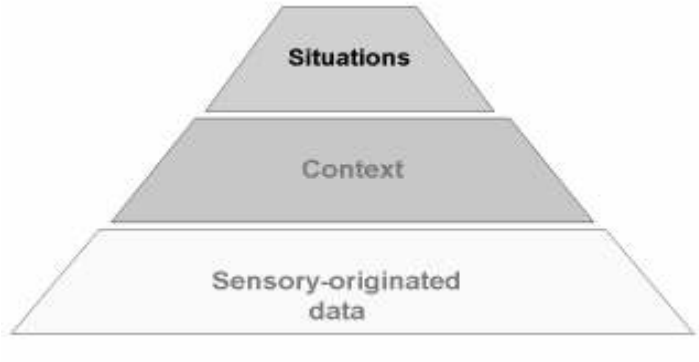


Figura 1: Pirámide de Situación de Contexto

4.3 Tipos de Contexto

Se diferencian principalmente cuatro tipos de contexto, que por su utilidad se destacan de los demás: Ubicación , Identidad, Actividad ,Tiempo. Con estos tipos de contexto no solo se responden las cuestiones de quién, qué, cuándo y dónde, sino que también se indican otras fuentes de información contextual. Por ejemplo si se tiene la identidad del usuario, seguramente se tendrá información secundaria como e-mail, número telefónico, edad, lista de contactos, etc. En otras palabras se tiene un sistema de dos niveles en el cual las categorías Ubicación, Identidad, Actividad y Tiempo determinan el primer nivel y el segundo nivel esta compuesto por cualquier otro tipo de información contextual. Las piezas de información secundaria tienen como característica que puede ubicar en un espacio de información cuyo índice es uno o varios de los tipos de contexto del primer nivel. Por ejemplo los números telefónicos de los usuarios pueden encontrarse indexados por la identidad del usuario [25].

Esta caracterización ayuda a los diseñadores a seleccionar el contexto a utilizar en sus aplicaciones, estructurar la forma de uso del contexto, y explorar otros contextos relevantes. Los cuatro primeras tipos de contextos indican los tipos de información necesarias para caracterizar una situación y el uso de las mismas provee una manera de usar y organizar el contexto [3]. Ver figura 2.
Un sistema es context-aware si el mismo usa el contexto para proveer información relevante y/o servicio al usuario, donde la relevancia depende de la tarea del usuario. Las definiciones de sistemas context-aware fueron elaboradas por Schilit [22], y luego continuadas por Dey [1],[9] entre otros. Las de éste último autor son las que hoy en día prevalecen y son más ampliamente aceptadas. Se detalla en la tabla 1 las funciones de Contex-Aware.

Tabla 1 Funciones de Contex-Aware

Funciones de Context-Aware	
Presentación de información y servicios:	Referida a aplicaciones que presentan información de contexto al usuario, Como ejemplo se puede citar una aplicación para teléfonos celulares que en función de la ubicación del usuario dentro de una ciudad le presenta los restaurantes más próximos.
Ejecución Automática de servicios:	Describe aplicaciones que disparan un servicio, o reconfiguran su comportamiento de acuerdo al cambio de contexto del usuario. Como ejemplo, la aplicación que incremente automáticamente el volumen y brillo del celular en los horarios en que una persona viaja a su trabajo.
Adjuntar información de contexto para su posterior recuperación:	Incluye aquellas aplicaciones que pueden poner en una etiqueta (TAG virtual), datos con información relevante del contexto, dejándolas

	para un posterior uso.

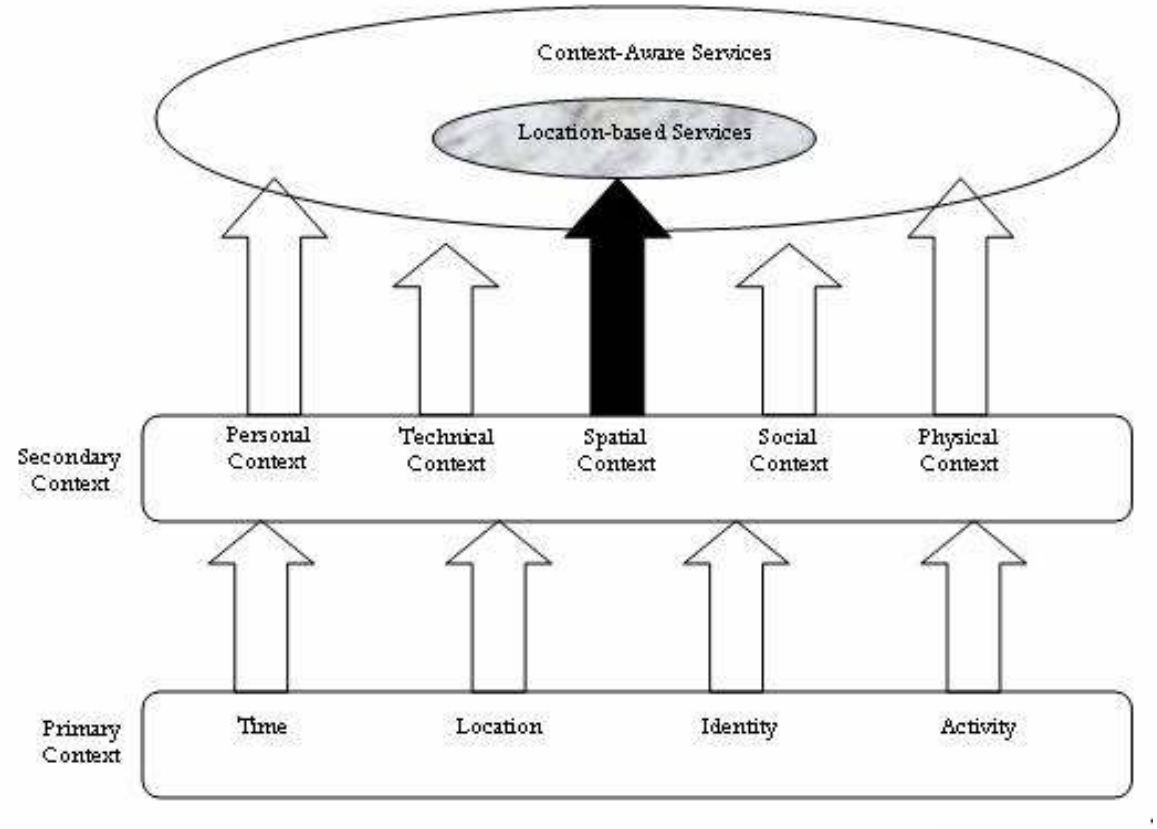


Figura 2: Tipos de contexto y Servicios Basados en Ubicación

5. SOLUCIONES

5.1 Introducción

Un tema muy interesante que se esta desarrollando en la actualidad es la integración de sistemas móviles a la vida diaria y crear escenarios en los que todos los sistemas capaces tengan la posibilidad de hablar con otros y prestarse servicios de distinta índole, ya sea intercambio de tarjetas de contacto, archivos, pago de cuentas, localización de personas, adaptación automática del software, etc. La frase “la computación del siglo 21” esta cada vez mas cerca para orientar a que la tecnología debe dejar de ser explicita a nosotros y volverse cada día mas integrada, pero para ello hay varias corrientes tratando de abordar el problema. El proyecto Kapsch propone la creación de un sistema de control del tránsito, sustentado en la unificación del cobro de peaje electrónico y en el montaje de una base operativa encargada del ordenamiento del tráfico y la seguridad vial. La iniciativa privada apunta a reemplazar los sistemas de telepeaje de los accesos por un nuevo dispositivo similar a un código de barras, que permitirá pagar los peajes a 60 kilómetros por hora. Ver figura 3 .Las principales funcionalidades que brinda este proyecto son:

- Control de señalización de las vías a través de carteles de mensajería variable, informando a los usuarios el estado de la circulación.
- Realizar mediciones y control de las condiciones ambientales en los túneles.
- Detectar y extinguir incendios.
- Atender llamadas de emergencia de los usuarios.
- Reproducir mensajes de voz, a través de altoparlantes, indicando a los usuarios las medidas y/o acciones a tomar en caso de incidente.
- Supervisión visual y detección de incidentes mediante video.
- Alimentar los sistemas de emergencia en caso de cortes en el suministro eléctrico.
- Monitorear en tiempo real secciones enteras de la autopista a través de cámaras y sensores.
- Ajustar de modo automático la iluminación al interior del túnel, considerando las condiciones de iluminación exterior.
- Ventilar automáticamente y mantener los niveles de CO y opacidad del aire en niveles de acuerdo a normativas y recomendaciones medioambientales.
- Detectar y extinguir posibles focos de incendio.
- Gestión de despacho de los recursos.

Sin embargo en este proyecto no se tiene en cuenta la posibilidad del uso de información contextual, ya que los usuarios del sistema (conductores), contienen en si este tipo de información que podría ser utilizado para mejorar la dinámica del sistema. Los usuarios no tienen acceso en todo momento a la información del contexto en donde están interactuando, solo a través de eventuales sistemas de información al paso como los carteles y altoparlantes, cuyo defecto es el de proporcionar ésta información una vez producido el hecho.

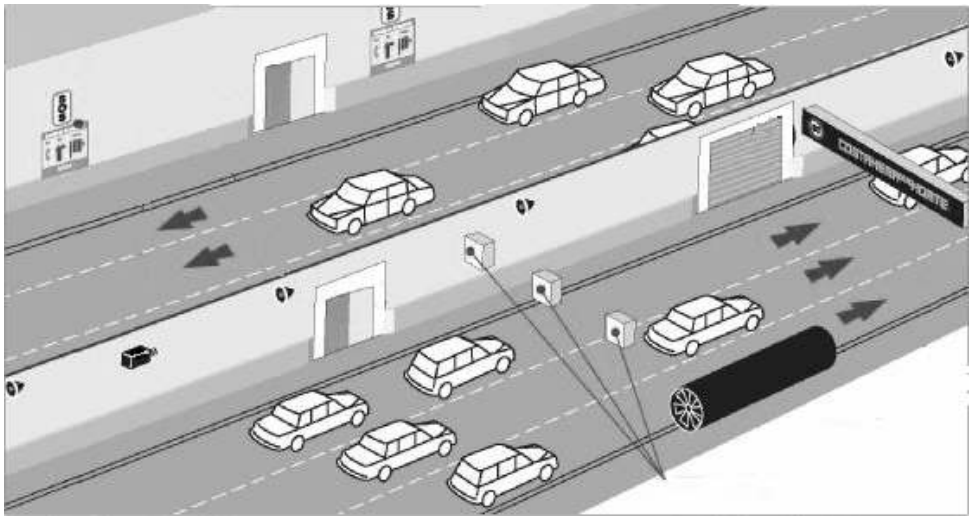


Figura 3: Sistema de Control de Autopistas HCS

5.2 Solución orientada a Context-Aware

La solución propuesta consiste en la utilización de aplicaciones context-aware ubicadas dentro de los vehículos y en las autopistas. Esta aplicación propuesta sería una aplicación móvil y la misma utilizaría servicios basados en localización (LBS), estos sistemas se encargan de proveer el posicionamiento geográfico del usuario. El contenido de un vehículo consiste principalmente en componentes mecánicos y electrónicos. Sin embargo el software es cada ves mas una pieza importante en los vehículos futuros. El software utilizado en los sistemas inteligentes de transporte (ITS) forma parte de la estrategia internacional de la seguridad del camino para mejorar la movilidad, la comodidad y la seguridad de los usuarios del mismo. El 95% de los accidentes se atribuyen al error del conductor. Los ITS y los sistemas de ayuda al conductor podrían reducir el error del mismo aumentando el conocimiento de información de seguridad crítica relevante en el contexto. Tal sistema centrado en el contexto es difícil de programar debido a la insuficiencia y a la complejidad de la información del ambiente.

5.2.1 Beneficios de esta propuesta

- Integrar la información tomada del entorno (Contexto) entre automóviles y la autopista
- Mejorar la llegada de información a los diferentes automovilistas
- Prevenir la congestiones de tránsito
- Disminuir y controlar posibles accidentes
- Aumentar la calidad de información
- Registrar mediciones de tráfico
- Aumentar la satisfacción en el manejo diario
- Innovar una nueva tecnología

5.2.2 Escenario Propuesto

El escenario propuesto incluye a los conductores (usuarios), como los principales receptores de la información contextual. Esta información proviene de varias fuentes, alguna de las cuales son los propios usuarios como consecuencia de ejecutar acciones por causa de haber recibido información desde la interfaz de la aplicación, provocando con sus decisiones nuevos datos de contextos. Podemos identificar los siguientes tipos de contexto según se muestran en la tabla 2:

Entidad	Contexto Primario	Contexto Secundario
Vehículo	Ubicación	Posición en el autopista (segmento)
	Identidad	Patente Conductor Marca / modelo Estado del vehículo
	Actividad	Información hacia el conductor Velocidad de circulación

		Cambios de estado en el vehículo
	Tiempo	Segmento atravesado en determinado momento
Autopista	Ubicación	No aplica
	Identidad	Nombre del autopista Segmento del autopista
	Actividad	Señalización del camino Información hacia los vehículos
	Tiempo	Cantidad de vehículos por segmento en un determinado tiempo
Medio	Ubicación	No aplica
	Identidad	Nombre del lugar
	Actividad	Visibilidad Estado del tiempo
	Tiempo	Hora y fecha Estación del año
Centro de Control	Ubicación	No aplica
	Identidad	Nombre del centro de control
	Actividad	Información hacia las autopistas Cierre de accesos y carriles
	Tiempo	Registros en la base de datos con información de sucesos con marca horaria

Tabla 2 Tipos de Contextos

6. EJEMPLO USANDO CONTEXT-AWARE

Como ejemplo citamos el caso del sistema Higway de Information Society Technologies (IST) [14]. Higway esta orientado a conductores y peatones europeos con servicios eSafety y la actual necesidad de interacción con multimedia (texto, audio, imágenes, vídeo en tiempo real, voz/gráficos) y los servicios basados en localización. Los servicios eSafety estarán basados en la combinación de mapas en tiempo real, tecnología móvil de banda ancha (UMTS 3G, Wi-Fi), los sistemas de posicionamiento, las herramientas 2D/3D y síntesis de voz / interfaces de reconocimiento de voz. El objetivo de Higway es ofrecer una seguridad más alta y servicios de valor agreado basados en ubicación, donde la interaccion entre la persona, el vehículo y la infraestructura de la información se tratan de una manera integrada.

7. CONCLUSIONES

Luego del análisis general planteado en el presente trabajo, podemos concluir en que el uso de la información que proporciona el contexto, resulta de interés y contribuye a resolver de manera mucho más eficiente el problema planteado, facilitando el desarrollo futuro de autopistas inteligentes que interactúan con los automóviles, en este caso agentes que negocian entre si bajo el contexto de reglas de seguridad, frente a la situación general del trafico que se desarrolla en la autopista con otros agentes, El desarrollo de sistemas centrados en contexto se adecua mucho mejor a los entornos móviles que un sistema del tipo tradicional como el que se plantea como propuesta de implementación en nuestras autopistas. En relación a las futuras líneas de investigación se orientan al desarrollo de un modelo centrado en contexto, sobre el cual actúan los automóviles en nuestro caso de estudio agentes que utilizaran la autopistas inteligente, que constituye el ambiente de negociación multiagente, para su comprobación se exploraran soluciones en el marco de los modelos de vida artificial.

8. REFERENCIAS

[1] A. K. Dey, 1999, G. D. Abowd - Towards a Better Understanding of Context and Context-Awareness.

[2] A. K. Dey, 2000, Tesis. Providing Architectural Support for Building Context-Aware Applications. College of Computing, Georgia Institute of Technology.

[3] Bill N. Schilit, Norman Adams, and Roy Want. 1994. Context-Aware Computing Applications

[4] Brown M., "Supporting user mobility", *In IFIP World Conference on Mobile Communications*, 1996, pages 69–77.

[5] Brown P. J., Bovey J. D., and Chen X., " Context-aware applications: from the laboratory to the marketplace", *IEEE Personal Communications*, October 1997, pages 58-64.

[6] Brown P.J., "The Stick-e Document: a Framework for Creating Context-Aware Applications", *Electronic Publishing*, 1996, pages 259-272.

[7] Brown P.J., "Triggering Information by Context", *Personal Technologies*, 1998 pages 1-9.

[8] Cooperstock J., Tanikoshi K., Beirne G., Narine T., Buxton W., "Evolution of a Reactive Environment", 1995.

[9] Dey A. K., "Providing architectural support for building contextaware applications", PhD thesis, Georgia Institute of Technology, 2000

[10] Dey A.K.," Context-Aware Computing: The Cyber Desk Project", *AAAI 1998 Spring Symposium on Intelligent Environments, Technical Report*, January 1998, pages 51-54.

[11] Elrod S., Hall G., Costanza R., Dixon M., "Responsive Office Environments", CACM, 1993.

[12] Franklin D. and Flachsbart J., "All gadget and no representation makes jack a dull environment", *In Proceedings of AAAI 1998 Spring Symposium on Intelligent Environments*, 1998.

[13] Hull R., Neaves P., Bedford-Roberts J., "Towards Situated Computing", *1st International Symposium on Wearable Computers*, 1997, pages 146-153.

[14] IST Project, 2003, Breakthrough intelligent maps & geographic tools for the context-aware delivery of e-safety & value-added services.

[15] Kortuem Gerd, Segall Zary, and Bauer Martin, "Context-aware, adaptive wearable computers as remote interfaces to 'intelligent' environments", *In ISWC*, 1998, pages 58–65.

[16] Pascoe J., "Adding Generic Contextual Capabilities to Wearable Computers", *2nd International Symposium on Wearable Computers*, 1998, pages 92-99.

[17] Pascoe J., Ryan N.S., Morse D.R., "Human-Computer-Giraffe Interaction – HCI in the Field", *Workshop on Human Computer Interaction with Mobile Devices,* 1998.

[18] Rekimoto J., Ayatsuka Y., Hayashi K., "Augment-able Reality: Situated Communication through Physical and Digital Spaces", *2nd International Symposium on Wearable Computers*, 1998.

[19] Ryan N., "Mobile Computing in a Fieldwork Environment: Metadata Elements", *Project working document,* 1997.

[20] Ryan N., Pascoe J., Morse D., "Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant", *Computer Applications in Archaeology*, 1997.

[21] Salber D., Dey A.K., Abowd G.D., "Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm", *Georgia Tech GVU Technical Report*, 1998.

[22] Schilit B., Theimer M., "Disseminating Active Map Information to Mobile Hosts", *IEEE Network*, 1994, pages 22-32

[23] Ward A., Jones A., and Hopper A., "A new location technique for the active office", 1997.

[24] Weiser Mark. 1991. The computer for the 21st century. Scientific American, 265(3):94—104

[25] Yorio Darío, 2006, Tesis. Identificación y Clasificación de Patrones en el Diseño de aplicaciones Móviles. Universidad Nacional de La Plata.

Aspectos de Gobernabilidad Electrónica aplicados a Sitios Web Municipales

Departamento de Ingeniería e Investigaciones Tecnológicas
Universidad Nacional de la Matanza
San Justo (1754) – Buenos Aires – Argentina

Daniel A. Giulianelli dgiulian@unlam.edu.ar
Rocío A. Rodríguez rrodri@unlam.edu.ar
Pablo M. Vera pablovera@unlam.edu.ar
Artemisa Trigueros artemisa@unlam.edu.ar
Isabel B. Marko marko@unlam.edu.ar
Mónica I. Larrosa mlarrosa@unlam.edu.ar

Abstract

Assuming that one of the aims of the municipality's web sites is to increase the quality of service offered to their citizens, this paper analyzes the aspects that will allow those sites to get a more effective approach to their citizens. Taking as a premise the intuition (as analogy of the subjects already known with those that will be known), this analysis wants to establish similarities and differences on web sites's design and functionality. National and international norms, regulations and publications are taken as a basis, in order to determine the most important aspects to perform the analysis. Finally, some aspects that have not been found in the quoted sources, were included, according to the needs, capabilities and restrictions of the studied communities. The relevance level of each aspect is determined from different points of view. The analysis of the web sites, considered in the investigation, allows watching if these sites fulfill the aspects. The use of automatic validation tools, internationally recognized, contributes with the evaluation of some of the aspect's fulfillment. This information allows building a weighed scale to determine the level of e-Government reached by each of the municipality's web sites which are the subject of this research.

Keywords: eGovernment, Municipality, Design, Friendly, Functionality, Accessibility, Usability.

Resumen

Asumiendo que uno de los objetivos de los sitios web municipales es incrementar la calidad de los servicios prestados a sus ciudadanos, este paper analiza los aspectos que permitirán a estos sitios, alcanzar un grado de acercamiento más efectivo con dichos ciudadanos. Tomando como premisa la intuición (como analogía entre lo ya conocido, con aquello próximo a conocer) este análisis busca establecer similitudes y diferencias en el diseño y funcionalidad de los sitios web. Normas, regulaciones y publicaciones nacionales e internacionales fueron tomadas como base para determinar los más importantes aspectos para realizar el análisis. Finalmente, algunos aspectos no encontrados en los sitios mencionados, fueron incluidos, de acuerdo a las necesidades, capacidades y restricciones de las comunidades estudiadas. El nivel de relevancia de cada aspecto está determinado desde diferentes puntos de vista. El análisis de los sitios web considerados en la investigación, permite observar en que medida cumplen con dichos aspectos. El uso de herramientas de validación automática, internacionalmente reconocidas, contribuye a la evaluación del cumplimiento de algunos de los aspectos considerados. Esta información permite construir una escala ponderada para determinar en que medida los sitios analizados favorecen el acceso de la comunidad a conceptos de Gobernabilidad Electrónica.

Palabras Claves: Gobernabilidad Electrónica, Municipios, Diseño, Amigabilidad, Funcionabilidad, Accesibilidad, Usabilidad

1 INTRODUCCIÓN

El desconocimiento tecnológico de gran parte de la población conforma una barrera muy difícil de superar cuando se mencionan términos tales como Conectividad, Internet, Banda ancha, entre otros. Muchas comunidades viven en un aislamiento tecnológico y continúan realizando innumerables gestiones en forma presencial que implican largas colas y viajes, ya que su desconocimiento conforma una barrera muy difícil de superar. Muchos de aquellos que se aventuran a las redes por curiosidad o motivación se encuentran con múltiples interfaces, conectividad de muy diversos tipos, páginas muy distintas y en muchos casos poco claras con lo cual su inquietud se transforma en desconcierto. En este entorno en Argentina se comienza a considerar la gobernabilidad electrónica como una herramienta que simplifique la vida de los ciudadanos.

Resultaría mucho más simple para aquellos que no están habituados a navegar por la Web, encontrar siempre los menúes en el mismo lugar, contar con un mapa de sitio para ver si la información que se requiere se encuentra o no alojada en dicha página entre otras características. Toda vez que los sitios cumplieran con un estándar sería más sencillo encontrar en ellos más rápidamente la información requerida. Los estándares de diseño posibilitan hacer una analogía entre los conocimientos adquiridos en el manejo de un software, con otro que se esté viendo por primera vez, permitiendo al usuario no experimentado tener conocimientos básicos de manejo inicial del software desconocido. La baja observancia de estándares en los sitios web, donde el afán por lo innovador ha dejado de lado la importancia de los mismos, dificulta el acceso a los ciudadanos. Los sitios gubernamentales deberían, tener una misma apariencia y cumplir con normativas que serán imprescindibles al momento de utilizar Internet para materializar de forma efectiva gobernabilidad electrónica.

Bajo nuestra premisa: "Cumplir con estándares de diseño, centrados en el usuario ayuda a la analogía de lo ya conocido con lo próximo a conocer, reduciendo los tiempos de búsqueda e incrementando la amigabilidad de un sitio web"**,** nos proponemos los siguientes objetivos:

- Centrar el estudio en los sitios web desarrollados por los municipios del conurbano bonaerense.
- Analizar los estándares existentes en Argentina y otros países.
- Analizar las páginas de los municipios a fin de poder establecer tópicos que mejoren la calidad de los sitios Web.
- Proponer una plantilla cuantificable sobre aspectos a tomar en cuenta en el diseño de sitios orientada a funcionalidad, amigabilidad, usabilidad, accesibilidad y veracidad.

2 EJE CENTRAL DEL TRABAJO

Nos proponemos evaluar en que grado los sitios web de los municipios del conurbano bonaerense (ver Figura 1[1]) cumplen con distintos aspectos contenidos en los conceptos básicos que interactúan para lograr la calidad del mismo.

[1] Mapa obtenido de: http://www.ic.gba.gov.ar/regionesculturales/r_conurbano.htm

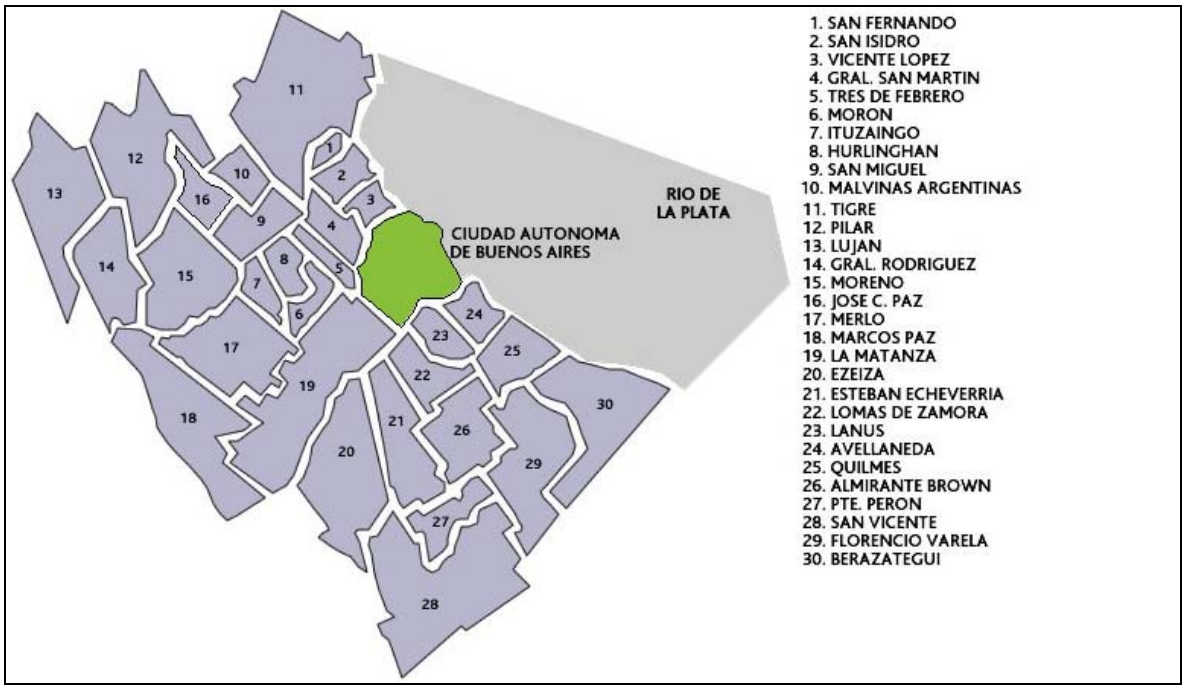


Figura 1: Municipios del Conurbano Bonaerense

Los conceptos que contribuyen a la calidad de un sitio web y que se han tomado en cuenta para realizar el análisis sobre los sitios objeto de estudio son los siguientes:

a. Funcionalidad: Entendemos por funcionalidad a la propiedad de facilitar al vecino del municipio el acceso a servicios tales como: información, trámites, pagos, estado de su deuda, etc., sin necesidad de presentarse personalmente en las oficinas municipales o locales de pago, permitiéndole ahorrar al vecino y al municipio, tiempo, dinero y esfuerzo. Por lo cual nuestra visión de los sitios web está orientada por el "Diseño Centrado en el Usuario".

b. Amigabilidad: El usuario debe poder fácilmente navegar por el sitio Web sin tomar en cuenta su nivel de experiencia. El lenguaje de comunicación debe resultar muy simple con un diseño que permita acceder a los contenidos más consultados de forma sencilla.

c. Usabilidad: Calidad de la experiencia del usuario cuando interactúa con un producto o sistema. La norma ISO 9241-11 [9] la define como: "La propiedad de un producto de ser usado por un usuario para alcanzar metas específicas con efectividad, eficiencia y satisfacción, dentro de un contexto de usuario particular". La norma ISO 13407[10] caracteriza al diseño centrado en el usuario mediante:
    - la participación activa del usuario y el claro conocimiento de los requerimientos usuario y tarea.
    - una apropiada localización de la función entre el usuario y la tecnología.
    - la iteración del diseño de soluciones.
    - el diseño multidisciplinario.

    La usabilidad puede ser medida mediante numerosos aspectos tales como: efectividad, eficiencia, satisfacción, facilidad para ser aprendido, permanencia en la memoria, frecuencia y severidad de los errores.

d. Accesibilidad: Es la posibilidad que un producto o servicio web pueda ser accedido y usado por el mayor número posible de personas, indiferentemente de las limitaciones propias del individuo o de las derivadas del contexto de uso.

e. Veracidad: Los sitios gubernamentales no deben tener aclaraciones de deslinde de responsabilidad sobre la información propia. Lo consignado debe ser de carácter informativo (actualizado, verídico y de relevancia para quién ingresa al sitio).

Tomando en cuenta estos criterios se evalúan que aspectos están contemplados por los estándares oficiales (nacionales e internacionales) que sean de aplicabilidad para analizar el cumplimiento de los mismos en los sitios web municipales. Con posterioridad se analizan publicaciones extranjeras a fin de tomar en cuenta aspectos no contemplados por las normativas precedentes y que tengan aplicación en los sitios de estudio. Concluido el análisis mencionado y en base a la experiencia aportada por la investigación, el equipo incorporó criterios de relevancia que deben ser tomados en cuenta en los sitios que nos ocupan (ver figura 2).

Figura 2: Basamento para analizar los aspectos que deberían considerar los sitios municipales

Los municipios de Ezeiza, Lanús, Marcos Paz, Merlo y San Vicente, que hasta el momento no cuentan con una página municipal oficial, han sido descartados del conjunto de las localidades a evaluar.

3 DEFICIENCIAS ENCONTRADAS

A continuación se presentan las principales falencias encontradas en los sitios analizados:

- Inexistencia de ayuda en línea (88%)
- No se cuenta con un mapa de ubicación geográfica (75%)
- Sin posibilidad de búsqueda de información interna del sitio (50%)
- Ausencia de un mapa de sitio (50%)
- Los datos municipales básicos para el contribuyente no se informan en el sitio (46%)
- Presentan solo búsquedas simples (39%)
- Ausencia de un formulario de contacto (38%)

Más información sobre deficiencias, porcentajes y comentarios puede consultarse en [6]

Al analizar los dominios de los sitios en estudio, se observa que el 65% de ellos tienen por dominio el nombre del municipio, lo cual permite al usuario acceder sin utilizar un buscador. Cabe destacar, que salvo en el caso de Luján en el cual el dominio “lujan.gov.ar” está destinado a presentar información turística y tiene enlace con el sitio de la municipalidad que se encuentra cargado como página secundaría dentro de dicho dominio, en el resto de los casos, los dominios del tipo

"NombreDeLaLocalidad.gov.ar" no son utilizados actualmente con lo cual al no estar reservados para otros fines podrían haberse elegido dichos dominios. La pregunta que queda entonces planteada es: ¿Por qué algunos municipios han optado por dominios tales como msm en vez de sanmiguel.gov.ar ó miituzaingo en vez de ituzaingo.gov.ar?

Dominios no concordantes con el nombre del partido: Avellaneda, Hurlingham, Ituzaingo, José Clemente Paz, Luján, Presidente Perón, San Miguel, Vicente López.

4 CUMPLIMIENTO DE LAS NORMATIVAS

4.1 W3C

W3C[2] (World Wide Web Consortium), es un consorcio internacional que tiene por objetivo llevar a Internet al máximo potencial desarrollando estándares y guías que aseguren el crecimiento a largo plazo de la web. Fue creado el 1º de octubre de 1994 por Tim Berners-Lee en el MIT (Massachusetts Institute of Technology que es una de las principales instituciones de Estados Unidos dedicadas a la docencia y a la investigación, especialmente en ciencia, ingeniería y economía). W3C propone todos los aspectos formales que son explicados en el apartado 4.2 (por surgir de la organización nacional que rige los sitios web argentinos) y agrega los siguientes tres aspectos: cumplimiento de HTML 4.01, accesibilidad desde dispositivos móviles y ayuda para manejo sin mouse (solo por medio de un teclado).

4.1.1 Cumplimiento de HTML 4.01

El W3C ha establecido como un aspecto relevante para los sitios web el cumplimiento del estándar de HTML. La totalidad de sitios web que satisfagan este aspecto podrán ser visualizados correctamente independientemente del navegador que se utilice. Podremos decir entonces que el sitio será Crossbrowsing.

Los errores implican definiciones de código no soportadas por el estándar de HTML 4.01 y han sido detectados por el validador de HTML 4.01 Estricto [19].

Se exhiben en la Tabla 1, los sitios con máxima y mínima cantidad de errores. Mayor información, incluyendo características del servidor donde se encuentran alojados, puede consultarse en [7].

Tabla 1 – Cantidad de errores de los Sitios Web Municipales por incumplimiento del estándar HTML 4.01

Municipalidad	**Cantidad de Errores** HTML 4.01 Strict
General Rodríguez	1063
Berazategui	3
Tigre	3

4.1.2 Accesibilidad desde dispositivos móviles

W3C ha establecido como otro aspecto relevante para los sitios web el nivel de accesibilidad de los sitios por parte de dispositivos móviles. W3C toma en cuenta para establecer el grado de accesibilidad de dispositivos móviles un muy importante número de principios. El equipo de

[2] Página principal de la W3C: http://www.w3.org/

investigación analizo los mismos desde el punto de vista del mercado local y la viabilidad de los habitantes del municipio y elaboró un resumen disponible en [4].

La visualización de los sitios fue testeada a través de los siguientes teléfonos celulares: Sony Ericsson W810, Sony Ericsson K750, Nokia N70.

De dicho análisis resultaron los siguientes niveles de cumplimiento para los sitios web analizados:

- Medio: Avellaneda, Berazategui, Esteban Echeverría, Florencio Varela, Gral. San Martín, José C. Paz, Pilar, Pte. Perón, San Miguel, Tigre, Tres de Febrero, Vicente López.
- Insuficiente: Alte. Brown, Gral. Rodríguez, Hurlingham, Ituzaingó, Lomas de Zamora, Luján, Malvinas Argentinas, Moreno, Morón, San Fernando, San Isidro.
- Bajo: La Matanza, Quilmas

4.2 ONTI

La Oficina Nacional de Tecnologías de Información (ONTI[3]) es el órgano rector en materia de empleo de tecnologías informáticas de la Administración Pública Nacional. Funciona en el ámbito de la Subsecretaría de la Gestión Pública de la Jefatura de Gabinete de Ministros y está formada por un grupo multidisciplinario de profesionales y especialistas en la materia. Tiene como objetivo formular políticas para la implementación de procesos de desarrollo e innovación tecnológica para la transformación y modernización del Estado, así como también propiciar la integración de nuevas tecnologías en el sector público, su compatibilidad, interoperabilidad y la promoción de la estandarización tecnológica. ONTI [12] consigna aspectos (que también son considerados a nivel internacional por la W3C), de riguroso cumplimiento para sitios web argentinos, comprendidos en las siguientes categorías: Banner, Contenido, Organigrama, Organización de Texto, Dominio, Menú principal, Categorías Accesibles desde el menú principal, Descargas, Documentos publicados, Enlaces a otros sitios, Estado de alta del sitio en los principales buscadores, Formularios, Imágenes y tamaño de páginas, Navegadores, Colores, Listas, Usabilidad, Cuestiones Legales. Mayor información sobre aspectos de riguroso cumplimiento valorizada esta disponible en [2].

Tomando en cuenta que se analizan sitios web de municipios del conurbano bonaerense y la experiencia de los miembros del grupo de investigación al respecto, tanto para los aspectos propuestos como para los obligatorios se ha establecido un índice de relevancia. El índice muestra la importancia del aspecto a través de una escala de 5 valores en donde el valor 5 representa la mayor relevancia, el valor 3 ha sido asignado a aquellos aspectos de relevancia Media y el valor 1 a los aspectos de mínima relevancia. Los valores intermedios (4 y 2) se han utilizado para brindar una mayor exactitud en la calificación de la relevancia del aspecto. Resulta importante dejar aclarado que la asignación del índice de relevancia está fundada en la experiencia, producto de la investigación llevada a cabo por los miembros del equipo. Disponible en [8].

ONTI también propone aspectos a modo de sugerencia de los cuales se han tomado trece (13) que son aplicables a los sitios web en estudio. El equipo de trabajo valorizó estas sugerencias según la escala anterior. Esta valoración puede ser consultada en [5].

Tomando en cuenta los 65 aspectos de riguroso cumplimiento enunciados por la ONTI y consignando una calificación binaria (SI – NO) para cada uno de ellos según las páginas de los sitios web municipales los contemplen o no (dejando de lado el grado de relevancia asignado a cada uno de ellos) y redondeando se ha elaborado para cada sitio municipal un porcentaje de grado de

[3] Página principal: http://www.sgp.gov.ar/contenidos/onti/onti.html

cumplimiento. Ejemplo: a un sitio que contempla 50 aspectos sobre un total de 65 se le ha asignado un porcentaje de cumplimiento del 77 % ya que (50 / 65) * 100) = 76,92.

En base a este criterio de cálculo, es posible señalar que ninguno de los sitios municipales investigados cumple en un 100 % con las especificaciones obligatorias establecidas por ONTI. Los extremos de cumplimiento de los aspectos obligatorios son los siguientes:

- Mayor cumplimiento : Municipio de MORON: 77%
- Menor cumplimiento : Municipios de TRES DE FEBRERO Y JOSE C. PAZ: 29%

5 PUBLICACIONES EXTRANJERAS

Tomando en cuenta publicaciones extranjeras ([1], [14], [22] [15], [13]) se consideran los aspectos que tienen aplicabilidad a los sitios web municipales del conurbano bonaerense y que no están contemplados por la ONTI ni por la W3C.

Tabla 2 – Aspectos establecidos en publicaciones extranjeras que pueden ser aplicados a los sitios web analizados.

AGREGADOS A LAS NORMATIVAS (ONTI y W3C) contemplados en publicaciones extranjeras	IMPOR-TANCIA	ORIGEN DEL ASPECTO			
		AUSTRALIA	CHILE	NUEVA ZELANDA	USA
Todas las funcionalidades necesarias están disponibles sin abandonar el sitio.	4	X	X		
Posee sección de "preguntas frecuentes" y sus respuestas	3				X
No utiliza marcos	5	X		X	X
Página con versión imprimible	4			X	
Propagandas - no permitir avisos publicitarios privados	5	X		X	X
Transporte para llegar a la municipalidad	4		X		

El equipo de investigación les asigna un valor de importancia a cada uno de ellos utilizando el mismo criterio que para los aspectos precedentes.

6 TRABAJOS RELACIONADOS

En el ámbito de habla hispana, dos estudios significativos son los realizados en España (Europa) y Chile (Latinoamérica). En el estudio español [16], el énfasis está puesto en la evaluación de los sitios web municipales tomando en cuenta el punto de vista de quién lo está utilizando (contemplando la posibilidad que éste sea o no experto). Además se considera el punto de vista técnico en cuanto a los costos evaluando entre el grado de dificultad para su automatización y su importancia en el ámbito específico de las sedes web municipales.

El estudio chileno [13], realiza su análisis evaluando el estado actual de los sitios web municipales, y estableciendo estrategias de corto y largo plazo para el manejo de gobernabilidad electrónica. También intenta potenciar mejores prácticas en los servicios ofrecidos a través de la web.

En ambos casos, estos estudios tienen puntos de contacto con el realizado en nuestra investigación. La diferencia reside por un lado en que nos centramos en las normativas nacionales e internacionales tomando en cuenta los aspectos propios para los sitios web. Por otra parte nuestro objetivo final será crear una propuesta de estandarización. Esta contribución tomará como base los resultados obtenidos en la presente investigación, la tarea desarrollada con los organismos municipales dispuestos a sumarse a la iniciativa y el aporte de organismos nacionales (ONTI).

7 OTROS ASPECTOS RELEVANTES

Para finalizar se consignan aspectos que a criterio de los miembros del equipo de trabajo tienen relevancia en los sitios web municipales del conurbano bonaerense. La tabla 3 muestra los aspectos y su importancia. Mayor información incluyendo comentarios se encuentra en [3]

Tabla 3 – Aspectos contemplados por el equipo de trabajo.

AGREGADOS A LAS NORMATIVAS contemplados por el equipo de trabajo	**IMPORTANCIA**
Diseño de las páginas secundarias concordantes con el de la principal.	5
En caso de tener página de presentación debería poderse saltear antes de la carga de la animación o imagen.	5
El sitio no debe tener música.	4
Los links deben estar subrayados	4
El color para los links ya visitados debe ser distinto al de los sin visitar.	4
Al posicionar el puntero del Mouse sobre una opción del menú o un link, se reemplaza el icono flecha por el icono de "la mano".	4
Permitir el uso del botón secundario del Mouse	3
En la barra de título del navegador se debe indicar el nombre de la página actual y el nombre del organismo (Ejemplo: Trámites-Mun.de Avellaneda)	3
En el menú principal cuando el usuario se posiciona sobre una opción, ésta debe resaltarse	3
En el resultado de los buscadores (por ejemplo: Google), debe aparecer Nombre de La Municipalidad, aclaración de que es el sitio oficial y país.	2
Páginas Principales centradas.	1
Icono en la barra de direcciones	1
Al posicionar el Mouse sobre un link éste cambia de color.	1

8 JERARQUIZACIÓN DE ASPECTOS

Tomando como base lo indicado por las normativas de los organismos: W3C y ONTI, junto con los aspectos propuestos por publicaciones extranjeras e incorporando la propuesta del grupo, se han enunciado 100 aspectos (78 ONTI + 3 W3C + 6 PUBLICACIONES EXTRANJERAS + 13 PROPUESTOS POR EL EQUIPO DE TRABAJO) que clasificamos mediante un nivel de importancia, como se detalló anteriormente quedando distribuidos como se muestra en la Tabla 4.

Tabla 4 – Propuesta del grupo de trabajo de jerarquización de aspectos.

INDISPENSABLES	Cantidad de Aspectos: 33
MUY IMPORTANTES	Cantidad de Aspectos: 26
IMPORTANTES	Cantidad de Aspectos: 28
DESEABLES	Cantidad de Aspectos: 8
OPTATIVOS	Cantidad de Aspectos: 5

8.1 Cumplimiento de Aspectos

El grupo de investigación chequeó el cumplimiento de cada uno de ellos en los sitios municipales del conurbano bonaerense. La Figura 3 muestra las áreas de los aspectos indispensables, muy importantes e importantes y la situación de los sitios web municipales. Esta gráfica resulta de asignarle el valor uno a los aspectos que fueron cumplimentados y cero en caso contrario, de modo que al realizar la suma de dichos valores se obtiene por resultado la cantidad de aspectos cumplimentados. Ejemplo: en el caso de Almte. Brown la cantidad de aspectos cumplimentados es:

Indispensables: 23, Muy Importantes: 11 e Importantes 8. Con lo cual la suma, por municipalidad, de las alturas de las tres áreas representadas en la Figura 3 aporta información sobre el cumplimiento de aspectos de mayor relevancia que conducirán, a criterio del equipo, más adecuadamente a facilitar la incorporación de los ciudadanos a Gobernabilidad Electrónica.

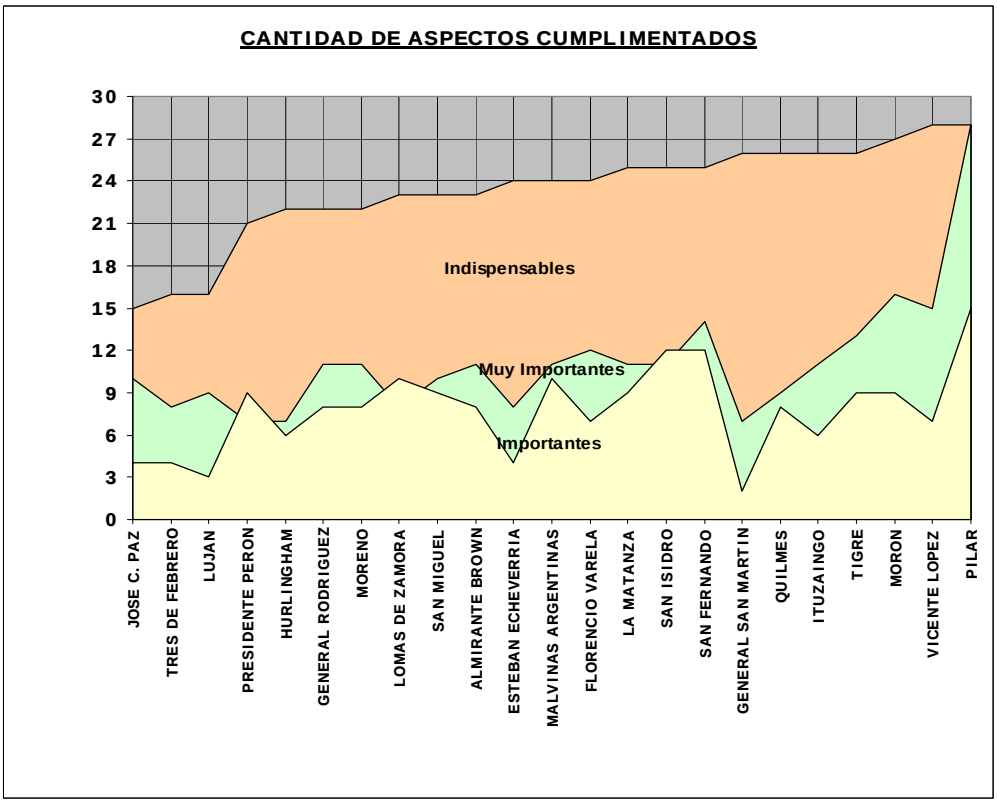


Figura 3 – Cumplimiento de los aspectos de mayor relevancia.

8.2 Cumplimiento de aspectos según nivel de importancia.

Ponderando por relevancia los aspectos, es posible asignar una puntuación basada en el cumplimiento de los mismos. La escala es la misma planteada a lo largo del paper. Esta gradación la hacemos coincidir con la propuesta que planteamos de jerarquías (ver Figura 3) de modo que un aspecto englobado en el nivel jerárquico más alto (ver tabla 5) merecerá una puntuación de 5. La numeración se asigna desde el nivel inferior hacia el superior en forma creciente, de esta forma un aspecto que se encuentre en el nivel más bajo tendrá asignado 1 punto. Por lo cual si se multiplica la cantidad de aspectos de cada nivel por su ponderación (número de nivel) esto dará la puntuación máxima que un sitio podría obtener (ver Tabla 5). Sumando las puntuaciones máximas por nivel, un sitio que cumple con todos los aspectos considerados, alcanza 374 puntos (suma de la última columna de la tabla 5).

Tabla 5 – Cálculo de puntuación por nivel

Nivel		Cantidad de Aspectos	Puntaje total posible de alcanzar por nivel
5	Indispensable	33	165
4	Muy Importante	26	104
3	Importante	28	84
2	Deseable	8	16
1	Optativo	5	5

Se calcula para cada municipio el cumplimiento de aspectos, de modo que se suma la puntuación de cada nivel. El puntaje de un nivel se obtiene contando la cantidad total de aspectos cumplimentados en dicho nivel, y al resultado se lo multiplica por el grado de relevancia del nivel. La Tabla 6 muestra el cálculo efectuado para el caso del municipio de Almirante Brown:

Tabla 6 – Cálculo de puntaje para el caso de Almirante Brown

Indispensable	23	x	5	=	115
Muy Importante	11	x	4	=	44
Importante	8	x	3	=	24
Deseable	1	x	2	=	2
Optativos	1	x	1	=	1
Puntaje					**186**

La Figura 5 muestra los puntajes obtenidos por cada uno de los sitios municipales relevados.

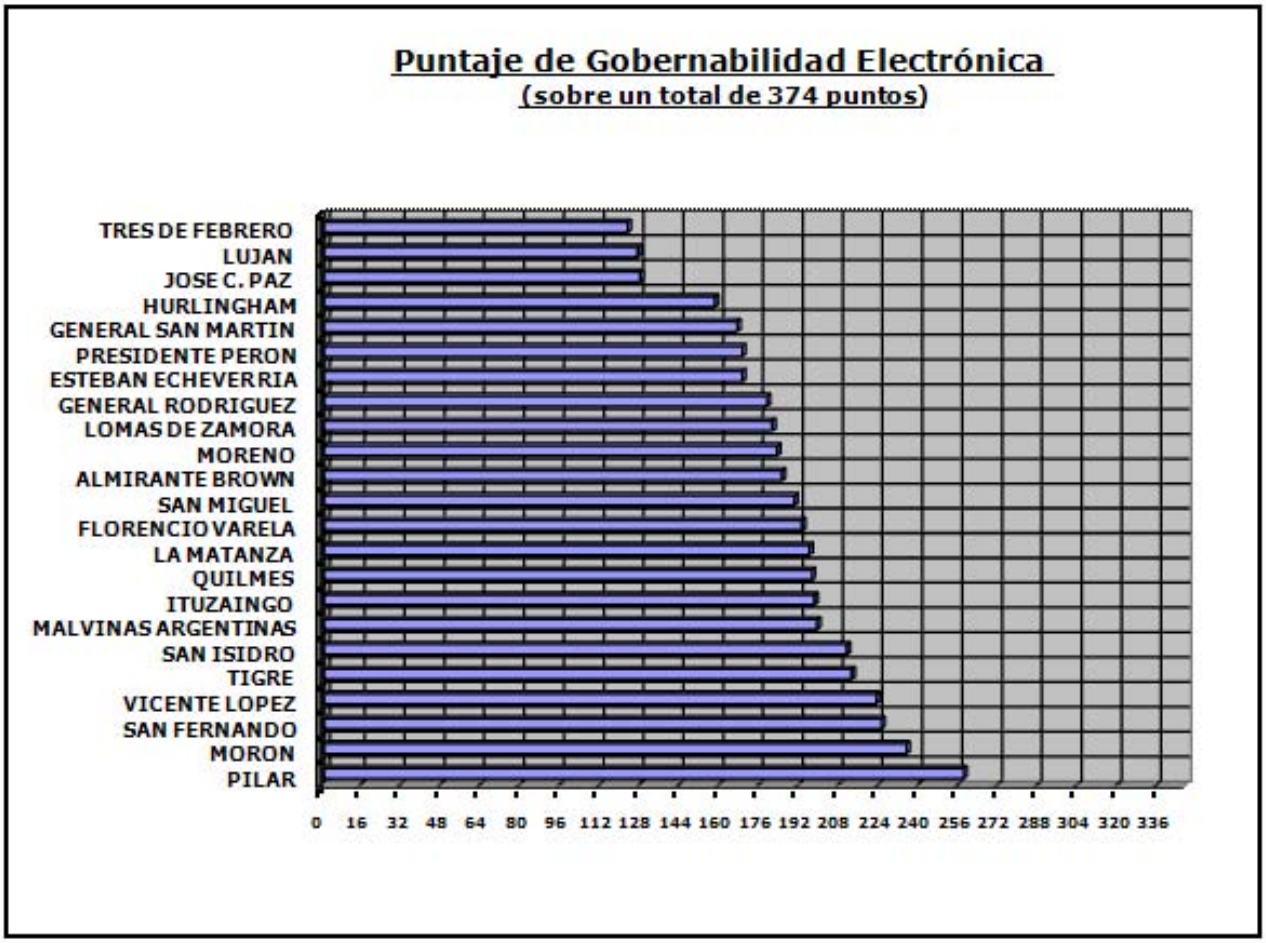


Figura 5 – Cumplimiento de aspectos generales (tomando en cuenta todas las jerarquías

9 CONCLUSIONES

A través del presente paper puede observarse que los sitios municipales del conurbano bonaerense no satisfacen totalmente lo establecido por las normas, publicaciones y propuestas que se tomaron como base. Hemos comprobado que no existe un patrón común para el diseño de los mismos. Los iconos utilizados para acciones comunes difieren (contacto, mapa de sitio, buscar...), así como la ubicación de los contenidos, etc. Las diferencias se aprecian tanto entre sitios distintos como en distintas páginas de un mismo sitio (el 21 % de la totalidad de sitios analizados presentan

variaciones de colores, tipografías, posición de los menues, etc.). Es muy frecuente observar sitios alojados en distintos servidores, desarrollados con tecnología diferente. Un muy significativo número de ellos no posee mapa de sitio e incluso algunos de esos sitios no brindan la posibilidad de realizar búsquedas con lo cual al usuario le puede resultar complicado recorrer todo el sitio para ver si encuentra en él la información deseada.

Habiendo relevado en cada uno de los sitios web municipales los 100 aspectos establecidos a la largo de este paper, no sólo pudimos analizar las diferencias y las principales falencias de los mismos sino que a su vez fue posible establecer una puntuación que indique la posición actual de cada uno de los sitios respecto a una serie de aspectos de gobernabilidad electrónica debidamente documentados. El 43% de los sitios no alcanzan a satisfacer el 50% de los aspectos que consideramos favorecen el acercamiento de los ciudadanos a la gobernabilidad electrónica.

El grupo de investigación considera necesario informar a los municipios el trabajo realizado y brindar, a aquellas entidades interesadas, la información básica en la que se fundamentan los resultados arrojados por los documentos de trabajo para que puedan tomar conocimiento de cuales son las causas que a nuestro entender, alejan a su comunidad de la gobernabilidad electrónica. Proponiendo hacer énfasis en mejorar prioritariamente el cumplimiento de aquellos aspectos con mayor nivel de relevancia.

10 CONTRIBUCIONES

- Informar a los distintos municipios su situación actual frente a los aspectos de gobernabilidad electrónica objetos de estudio
- Brindar apoyo en la medida que los municipios lo permitan para posicionar a los sitios web municipales en grados de cumplimiento más cercanos al ideal propuesto.
- Establecer un vínculo de colaboración con la Oficina Nacional de Tecnología de la información.

11 TRABAJOS FUTUROS

- Se prevé incorporar a los aspectos evaluados el uso de CSS[4], analizando si es utilizado por los sitios investigados. Este aspecto será validado a través del validador en línea de la W3C [18].
- Evaluar para cada uno de los aspectos su incidencia en funcionabilidad, amigabilidad, usabilidad, accesibilidad y veracidad de la información del sitio. Con esto será posible realizar un análisis de cómo esta posicionado cada sitio respecto a estos criterios y a través de una comparativa, analizar cuales son los sitios que cumplen en mayor medida con estas características.
- Extender este estudio hacia otros sitios municipales.
- Proponer a partir de los criterios considerados un estándar para sitios web municipales.

REFERENCIAS

[1] Department of the Premier and Cabinet - Office of e-Government; Guidelines for State Government Websites Version 2.1; Australia; 2006; http://www.egov.dpc.wa.gov.au/documents/WebGuidelinesVersion2.1_final.doc

[4] Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML)

[2] Giulianelli D, Rodríguez R, Vera P. y otros; Aspectos de cumplimiento obligatorio indicados por ONTI.; http://www.investigamos.com.ar/ge/obligatoriosonti.pdf

[3] Giulianelli D, Rodríguez R, Vera P.; Aspectos de Gobernabilidad Electrónica agregados por el grupo de investigación; http://www.investigamos.com.ar/ge/aspectosagregados.pdf

[4] Giulianelli D, Rodríguez R, Vera P. y otros; Aspectos para evaluar accesibilidad desde dispositivos móviles; http://www.investigamos.com.ar/ge/mobiledevices.pdf

[5] Giulianelli D, Rodríguez R, Vera P. y otros; Aspectos recomendados por ONTI; http://www.investigamos.com.ar/ge/sugeridosonti.pdf

[6] Giulianelli D, Rodríguez R, Vera P. y otros; Deficiencias encontradas en los sitios web municipales analizados; http://www.investigamos.com.ar/ge/deficiencias.pdf

[7] Giulianelli D, Rodríguez R, Vera P. y otros; Errores detectados de HTML 4.01 en los Sitios de Estudio; http://www.investigamos.com.ar/ge/erroreshtml.pdf

[8] Giulianelli D, Rodríguez R, Vera P. y otros; Fundamentos para determinar el grado de relevancia de cada aspecto; http://www.investigamos.com.ar/ge/aspectrelevance.pdf

[9] Manual de la norma ISO 9241-11
http://www.iso.org/iso/en/StandardsQueryFormHandler.StandardsQueryFormHandler?scope=CATALOGUE&sortOrder=ISO&committee=ALL&isoDocType=ALL&title=true&keyword=9241

[10] Manual de la norma ISO 13407
http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=21197&ICS1=13&ICS2=180&ICS3=

[11] ONTI; Plan de Gobierno Electrónico Decreto 378/2005;
http://www.sgp.gov.ar/contenidos/onti/productos/pnge/docs/pnge_decreto_378_2005.pdf

[12] ONTI; Sitios y Portales de Internet; Argentina;
http://www.sgp.gov.ar/contenidos/onti/etap/sitio_etap/docs_y_varios/Tecnologias/PaginasWeb.doc

[13] Pontificia Universidad Católica de Chile; Cátedra e-Government; Sitios Web Municipales e-Government Chile; 2006;
http://www.cetiuc.cl/wp-content/uploads/2007/01/presentacion-estudio-municipalidades.pdf

[14] State Services Commission; NZ Governmert Web Standards and Recommendations; Version 1.0; Nueva Zelanda; 2007;
http://www.e.govt.nz/standards/web-guidelines/web-standards-v1.0/web-standards-v1.0.rtf

[15] Subsecretaría de Desarrollo Regional y Administrativa; Ministerio del Interior; Gobierno de Chile; Resultados Segunda Encuesta Realidad Tecnológica Municipal; 2004;
http://www.subdere.gov.cl/1510/articles-68168_recurso_1.ppt

[16] Universidad Pompeu Fabra; Evaluación de la usabilidad en sistemas web municipales, metodología de análisis y desarrollo; Barcelona; 2005;
http://www.semanticaweb.net/archives/2005_evaluacion-municipales-isko.pdf

[17] W3C; Guía Breve de Accesibilidad Web;
http://www.w3c.es/divulgacion/guiasbreves/Accesibilidad

[18] W3C; Herramienta de validación de CSS; http://jigsaw.w3.org/css-validator/

[19] W3C; Herramienta de validación de HTML 4.01 estricto; http://validator.w3.org/

[20] W3C; Herramienta para el chequeo de links; http://validator.w3.org/checklink

[21] W3C; Mobile Web Best Practices 1.0 - Basic Guidelines;
http://www.w3.org/TR/2006/CR-mobile-bp-20060627/

[22] Web Managers; Requirements and Best Practices Checklist for Government; USA; 2005;
http://www.usa.gov/webcontent/reqs_bestpractices/checklist/long.pdf

Automatización y Gestión de las Pruebas Funcionales usando Herramientas Open Source

Ignacio Esmite, Mauricio Farías, Nicolás Farías, Beatriz Pérez
Centro de Ensayos de Software (CES), Universidad de la República
Montevideo, Uruguay, 11000
{iesmite, mfarias, nfarias, bperez}@fing.edu.uy

Resumen

En este artículo se presenta una metodología y el conjunto de herramientas open source utilizado para la automatización de las pruebas funcionales de productos con interfaz web. Este conjunto de herramientas está compuesto por: Selenium, Eclipse y extensiones de Mozilla Firefox como son Firebug, XPath Checker y XPather. Se describe la experiencia de utilizar la metodología en un proyecto de automatización específico y se concluye la factibilidad para la automatización de las pruebas siguiendo las actividades y el conjunto de herramientas definidos. Si bien las herramientas asisten en las pruebas automatizadas, no brindan soporte para la organización de los artefactos del proyecto: scripts, documentos y reportes de ejecución. Como trabajo a futuro se propone integrar al conjunto de herramientas definido, la herramienta FitNesse para gestionar los artefactos, buscando mejorar la organización de las pruebas junto con la comunicación y colaboración del equipo de pruebas.

Palabras claves: Ingeniería de software, Pruebas, Pruebas Funcionales, Pruebas Funcionales Automatizadas, Herramientas, Open Source**.**

1 INTRODUCCIÓN

La automatización de las pruebas funcionales reduce significativamente el esfuerzo dedicado a las pruebas de regresión en productos que se encuentran en continuo mantenimiento. La automatización de las pruebas debe ser considerada un proyecto en sí mismo con objetivos definidos.
En este artículo se presenta una metodología y el conjunto de herramientas open source utilizado para la automatización de las pruebas funcionales en productos con interfaz Web. Este conjunto de herramientas está compuesto por: Selenium[22], Eclipse[8] y extensiones de Mozilla Firefox[17] como son Firebug[9], XPath Checker[29] y XPather[30]. Se describe la experiencia de utilizar la metodología en un proyecto de prueba específico y se concluye la factibilidad para la automatización de las pruebas siguiendo las actividades y el conjunto de herramientas definidos. Si bien las herramientas asisten en las pruebas automatizadas, no brindan soporte para la organización de los artefactos del proyecto: scripts, documentos y reportes de ejecución. Como trabajo a futuro se propone integrar al conjunto de herramientas definido, la herramienta FitNesse[10] para gestionar los artefactos, buscando mejorar la organización de las pruebas junto con la comunicación y colaboración del equipo de pruebas.
En la sección 1.1 se describen los principales conceptos relacionados con las pruebas, en la sección 1.2 se introducen los conceptos de herramientas de automatización. Luego en la sección 1.3 se describe el contexto y motivación para este trabajo.
En la sección 2 se describe la experiencia práctica de automatizar las pruebas usando Selenium y otras herramientas open source.
En la sección 3 se muestra una propuesta para gestionar los productos resultantes de la automatización utilizando la herramienta FitNesse. Por último, en la sección 4 se presentan las conclusiones del artículo.

1.1 Principales Conceptos

Se presentan los principales conceptos relacionados con la automatización de las pruebas funcionales. Se define prueba funcional, caso de prueba, procedimiento de prueba, script de prueba, suite de prueba, pruebas de regresión y pruebas de humo.

El objetivo de la prueba funcional es validar cuando el comportamiento observado del software probado cumple o no con sus especificaciones. La prueba funcional toma el punto de vista del usuario [2]. Las funciones son probadas ingresando las entradas y examinando las salidas. La estructura interna del programa raramente es considerada [16].
Para realizar pruebas funcionales, la especificación se analiza para derivar los casos de prueba. Técnicas como partición de equivalencia, análisis del valor límite, grafo causa-efecto y conjetura de errores son especialmente pertinentes para las pruebas funcionales. Se deben considerar condiciones inválidas e inesperadas de la entrada y tener en cuenta que la definición del resultado esperado es una parte vital de un caso de prueba. El propósito de la prueba funcional es mostrar discrepancias con la especificación y no demostrar que el programa cumple su especificación [18].

Un caso de prueba (test case) es un conjunto de valores de entrada, precondiciones de ejecución, resultados esperados y poscondiciones de ejecución, desarrollados con un objetivo particular o condición de prueba, tal como ejercitar un camino de un programa particular o para verificar que se cumple un requerimiento específico [12].

Un script de prueba (test script) son los datos y las instrucciones escritas con una sintaxis formal, almacenado en un archivo y usado por una herramienta de automatización de las pruebas. Un script de prueba puede automatizar uno o más casos de prueba, navegación, inicialización u operaciones de configuración del entorno. Un script de prueba previsto para la ejecución manual de las pruebas es un procedimiento de prueba [26].
Una suite de pruebas es uno o más conjuntos de pruebas reunidos para satisfacer un objetivo de prueba [26]. Un conjunto de prueba incluye scripts y documentación. En nuestro caso, las suites constituyen un conjunto de scripts y el orden de ejecución de los mismos.

Las pruebas de regresión tienen como objetivo verificar que no ocurrió una regresión en la calidad del producto luego de un cambio, asegurándose que los cambios no introducen un comportamiento no deseado o errores adicionales. Implican la reejecución de alguna o todas las pruebas realizadas anteriormente [3].
Las pruebas de humo son un conjunto de pruebas aplicadas a cada nueva versión, su objetivo es validar que las funcionalidades básicas de la versión se comportan según lo especificado. Estas pruebas buscan grandes inestabilidades o elementos clave faltantes o defectuosos, que hacen imposible realizar las pruebas como fueron planificadas para la versión. Si la versión no pasa las pruebas de humo, no se comienza la ejecución de las pruebas planificadas de la versión [15].

1.2 Automatización de las pruebas

Hay herramientas que apoyan diversos aspectos de la prueba. A continuación se presenta una clasificación posible para las herramientas [13]:

- Administración de las pruebas y el proceso de pruebas: herramientas para la administración de las pruebas, para el seguimiento de incidentes, para la gestión de la configuración y para la administración de requerimientos.
- Pruebas estáticas: herramientas para apoyar el proceso de revisión, herramientas para el análisis estático y herramientas de modelado.
- Especificación de las pruebas: herramientas para el diseño de las pruebas y para la preparación de datos de prueba.
- Ejecución de las pruebas: herramientas de ejecución de casos de prueba, herramientas de pruebas unitarias, comparadores, herramientas de medición del cubrimiento, herramientas de seguridad.
- Desempeño y monitorización: herramientas de análisis dinámico, herramientas de desempeño, de carga y de estrés, herramientas de monitorización

Para la automatización de las pruebas funcionales son especialmente indicadas las herramientas de ejecución de las pruebas de captura y reproducción. Estas herramientas permiten al tester capturar y grabar pruebas, para luego editarlas, modificarlas y reproducirlas en distintos entornos. Herramientas que graban la interfaz de usuario a nivel de componentes y no de bitmaps son más útiles. Durante la grabación se capturan las acciones realizadas por el tester, creando automáticamente un script en algún lenguaje de alto nivel. Luego el tester modifica el script para crear una prueba reusable y mantenible. Este script se vuelve la línea base y luego es reproducido en una nueva versión, contra la cual es comparado. En general estas herramientas vienen acompañadas de un comparador, que compara automáticamente la salida en el momento de ejecutar el script con la salida grabada [7].

1.3 Contexto y Motivación

El Centro de Ensayos de Software (CES) [5] es un emprendimiento conjunto de la Universidad de la República de Uruguay (UdelaR) [28] y de la Cámara Uruguaya de Tecnologías de la Información (CUTI)[4], entidad que agrupa a la mayoría de las empresas productoras de software del país. Los servicios que ofrece el CES incluyen

- Servicios de prueba independiente: Planificar, diseñar, coordinar y ejecutar pruebas de productos de software de manera efectiva y controlada, definiendo claramente el contexto y los objetivos.
- Consultoría: Asesorar a las organizaciones en la mejora de los procesos de prueba, definición de estrategias y automatización de las pruebas. Colaborar en la creación y consolidación de sus áreas de prueba.
- Capacitación: Elaborar e impartir programas de capacitación en la disciplina de testing según las necesidades de cada organización.

Alguno de los clientes del CES, son empresas uruguayas medianas y pequeñas productoras de tecnologías de la información que muestran gran interés en automatizar sus pruebas funcionales. Gran parte de estas empresas comercializan uno o dos productos de software, los cuales personalizan para su venta en distintos clientes, debido a esto, los productos están en continuo mantenimiento y mejora. Con cada nueva versión del producto, las empresas necesitan asegurarse que las principales funcionalidades del producto continúan operando correctamente. Las empresas ven la importancia de utilizar la automatización para reducir los costos y tiempos en las pruebas de regresión y poder contar con un conjunto de pruebas de humo automatizadas. Muchas de ellas, buscan que el costo de la licencia no sea una barrera para comenzar el proceso de automatización. Esta es una de las razones que hacen más atractiva para ellas la automatización con herramientas open source.
Al momento de seleccionar una herramienta para las pruebas, las empresas buscan aquella que con el menor costo, les permita construir las pruebas automatizadas que mejor se adapten a su producto. La confianza en la herramienta cumple un papel fundamental. Para esto no solamente requieren tener personal con experticia en la herramienta, también tienen que asegurarse de que la misma podrá ser extendida para resolver nuevos problemas en caso de ser necesario. Para ello es muy importante evaluar el uso, la información y el soporte que tiene la herramienta en la comunidad.

2 AUTOMATIZACIÓN DE LAS PRUEBAS FUNCIONALES

Existe un proceso definido para las pruebas funcionales manuales en el Centro de Ensayos de Software. Dicho proceso cuenta con etapas, actividades, roles y artefactos definidos para un proyecto de prueba independiente, donde el equipo de pruebas es contratado para realizar las pruebas de un producto de software desarrollado por terceros. El proceso utilizado se llama ProTest y puede ser consultado en [21].
La experiencia en proyectos de automatización de pruebas funcionales motivó la necesidad de extender el proceso definido, con nuevas actividades específicas para la automatización.
En la sección 2.1 se presentan las actividades específicas definidas para la automatización de las pruebas. En la sección 2.2 se describen las herramientas open source utilizadas para dicha automatización, en la sección 2.3 se describe la experiencia en el CES de realizar las actividades con estas herramientas. Por último en la sección 2.4 se presentan las conclusiones sobre la utilización de herramientas open source para la automatización de las pruebas funcionales de productos de software.

2.1 Metodología Propuesta

A continuación se describen las actividades que utiliza el CES en sus proyectos de automatización. Se especifican los objetivos y se detallan las tareas realizadas en cada actividad. En la Figura 1 se muestra el diagrama de actividad para las actividades definidas y en la Tabla 1 se resumen los roles involucrados y los artefactos de entrada y de salida de cada actividad.

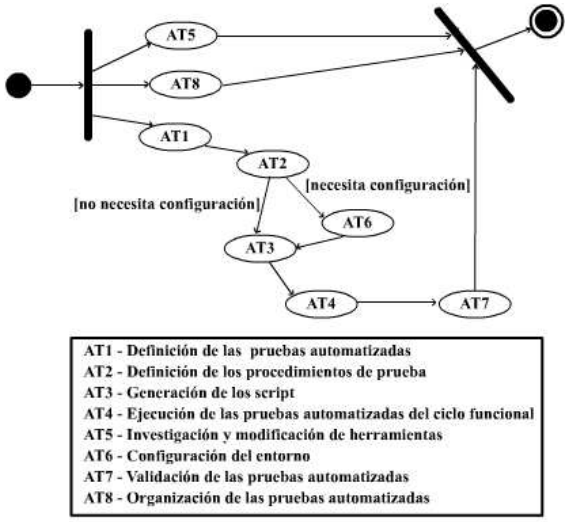


Figura 1 – Diagrama de Actividad

AT1 – Definición de las pruebas automatizadas
El objetivo de esta actividad es definir el conjunto de ciclos funcionales o funcionalidades que se probarán con las pruebas automatizadas.
Esta actividad consiste en definir con el Cliente qué pruebas se van a automatizar (en alto nivel), considerando los ciclos funcionales o funcionalidades que involucran. Para ello se priorizan y evalúan las pruebas y las funcionalidades a probar.

AT2 – Definición de los Procedimientos de Prueba
El objetivo de esta actividad es definir las suites y scripts que conformarán las pruebas automatizadas.
A partir de los ciclos funcionales y funcionalidades seleccionados para las pruebas, se especifican las suites y scripts que las ejecutarán. Definir las suites implica definir los scripts que las componen y especificar posibles dependencias de ejecución entre ellas. Para cada script se debe definir la tarea que debe realizar y las verificaciones que debe contener.

AT3 – Generación de suites y scripts
El objetivo de esta actividad es obtener las suites con sus scripts correspondientes.
Esta actividad consiste en el armado de cada una de las suites correspondientes al ciclo funcional a automatizar. Se graba o codifica las pruebas obteniendo como resultado los scripts integrantes de las suites. Se verifica el correcto funcionamiento de cada script.

AT4 – Ejecución de las pruebas automatizadas del ciclo funcional
El objetivo de esta actividad es ejecutar una prueba completa de las suites correspondientes al ciclo funcional y verificar su correcto funcionamiento.

Esta actividad consiste en realizar la prueba completa del ciclo funcional en el entorno preparado para dicho fin. En caso de un funcionamiento incorrecto deben realizarse los ajustes necesarios. En esta actividad se verifica el comportamiento de las suites en su conjunto.

AT5 – Investigación y modificación de herramientas
El objetivo de esta actividad es encontrar soluciones a las necesidades que no pueden satisfacerse con las herramientas de automatización que se manejan.
Esta actividad incluye recorrer foros, referencias y buscar antecedentes similares a la necesidad planteada. Luego se analizan las posibles soluciones. Estas incluyen instalar nuevas versiones de las herramientas, modificar las herramientas o extenderlas. Esta actividad también incluye búsqueda de nuevas herramientas que asistan en la automatización.

AT6 – Configuración del entorno
El objetivo de esta actividad es configurar el entorno de la aplicación que se desea probar para poder ejecutar las suites correctamente y documentar esta configuración.
En esta actividad se debe configurar los datos que mantiene la aplicación de manera que permitan la correcta ejecución de las suites. Es importante documentar con el detalle suficiente esta configuración.

ACTIVIDAD	ROLES	ENTRADA	SALIDA
AT1 - Definición de las pruebas automatizadas	Líder Diseñador de Pruebas Cliente	Requerimientos Acta reunión cliente	Documento que especifica las funcionalidades y ciclos funcionales a probar con las pruebas automatizadas
AT2 - Definición de los procedimientos de prueba	Diseñador de Pruebas Cliente	Requerimientos Acta reunión cliente Documento obtenido en AT1	Documento con Suites y Scripts que componen las pruebas
AT3 - Generación de suites y scripts	Tester	Documento obtenido en AT2 y documento obtenido en AT6	Suites y Scripts que las componen
AT4 - Ejecución de las pruebas automatizadas del ciclo funcional	Tester	Suites y Scripts obtenidos en AT3	Suites y Scripts verificados
AT5 - Investigación y modificación de herramientas	Tester	Herramienta a investigar	Nuevas herramientas (adquiridas o modificadas) y documentación pertinente
AT6 - Configuración del entorno	Tester	Aplicaciones a probar y documento obtenido en AT2	Aplicaciones aptas para ser probadas y documentación pertinente
AT7 - Validación de las pruebas automatizadas	Cliente Tester	Suites y Scripts obtenidos en AT4	Suites y Scripts verificados y validados
AT8 - Organización de las pruebas automatizadas	Diseñador de Pruebas Tester	Artefactos generados en el proyecto	Artefactos gestionados

Tabla 1 – Actividades, Roles y Artefactos de Entrada y de Salida

AT7- Validación de las pruebas automatizadas
El objetivo de esta actividad es verificar el correcto comportamiento de los scripts, en el ambiente de prueba del cliente y preparar las suites y scripts generados para la validación del cliente.
En esta actividad se prueba y eventualmente se ajustan los scripts para el correcto funcionamiento en el ambiente de pruebas del cliente. El cliente valida comparando las pruebas que espera automatizar con las pruebas que los scripts realizan.

AT8 – Organización de las pruebas automatizadas
El objetivo de esta actividad es gestionar los artefactos generados en el proyecto de automatización.
Esta actividad consiste en definir, actualizar y ejecutar los procedimientos para la gestión de los documentos, suites y scripts que se generen en el proyecto de automatización.

2.2 Herramientas utilizadas

El grupo de herramientas Selenium que ayudan en la ejecución de pruebas, está constituido por Selenium Core[23], Selenium IDE[24] y Selenium Remote Control[25]. Estas herramientas permiten crear y ejecutar pruebas automatizadas sobre aplicaciones Web. Las pruebas automatizadas pueden ser utilizadas como pruebas de regresión o para probar la aplicación sobre diferentes plataformas y navegadores Web. Cada una de estas herramientas son el producto de proyectos que continúan en desarrollo. Estos son llevados a cabo por la Comunidad Open QA[20] y utilizan la licencia Apache 2.0 [1]
La herramienta principal de Selenium es Selenium Core. Es una herramienta de automatización desarrollada inicialmente por programadores y testers de ThoughtWorks [27] y actualmente por OpenQA. Es una aplicación Web desarrollada con HTML y Javascript que soporta una gran variedad de plataformas y navegadores. Permite ejecutar una prueba o conjunto de pruebas automatizadas.
Selenium IDE es una herramienta de grabación que permite grabar pruebas para ejecutarlas en Selenium Core y se integra como una extensión en Mozilla Firefox . Registra las acciones que ejecuta el usuario a través del navegador, generando un script de prueba. Luego permite editarlo.
Una alternativa a Selenium Core es Selenium Remote Control. Esta herramienta de automatización brinda interfaces de programación (APIs) para lenguajes como Java, .NET, Perl, Python y Ruby. A través de ellas los programas pueden interactuar directamente con la interfaz Web de la aplicación utilizando navegadores como Mozilla Firefox e Internet Explorer[14].
Como entorno de desarrollo de los scripts se utilizó Eclipse, que brinda facilidad para crear y editar los scripts utilizando *plug-ins* para edición de HTML y XML. Además permite manejar el control de versiones de forma integrada, ofreciendo un cliente para conectar con el repositorio de scripts (CVS[6]). Cabe destacar que se hace uso de un conjunto mínimo de las funcionalidades que este entorno de desarrollo provee.
El navegador Web Mozilla Firefox permite integrar varias aplicaciones que ayudan a la creación de scripts. Entre ellas destacamos Firebug, XPath Checker y XPather.
La aplicación Firebug pone a disposición herramientas que permiten visualizar y editar fácilmente CSS, HTML y Javascript de la página en la cual se navega. También se puede ejecutar paso a paso el código Javascript de la página.
Por otro lado, XPath Checker y XPather identifican y muestran elementos de la página a través de su XPath. También pueden utilizarse de modo inverso, haciendo clic derecho sobre los elementos de la pantalla para poder obtener su XPath.

2.3 Experiencia

Desde que el CES inició la investigación en el área de automatización de pruebas funcionales, hace un año atrás, se han realizado varios proyectos donde se han automatizado pruebas funcionales para aplicaciones Web. Una de estas aplicaciones, desarrollada con Genexus [11], fue probada tanto en su versión para la plataforma Java como para la plataforma .NET. Otra de las aplicaciones para las cuales se automatizaron pruebas fue construida utilizando Java. En general, con mayor y menor dificultad se pudo lograr la automatización.
En el proyecto de automatización de las pruebas para la aplicación desarrollada con Genexus, el equipo de automatización que llevo a cabo este proyecto estuvo conformado por tres testers automatizadores, que cumplían también el rol de diseñador de pruebas y un líder de proyecto. El cliente quería automatizar un subconjunto de pruebas de regresión, para ejercitar los caminos y ciclos funcionales típicos de la aplicación. Se definieron dos etapas para la automatización, donde se automatizaron distintos escenarios de prueba. En la etapa 2 se extendieron las pruebas de la etapa 1 y se crearon pruebas para nuevos escenarios. Las dos etapas duraron dos y tres meses

respectivamente, en cada una de estas etapas se trabajó aplicando la metodología propuesta. En la tabla 2 se puede observar la cantidad de suites y scripts generados en cada etapa. Si apreciamos la relación entre la cantidad de suites y scripts generados y el tiempo de cada etapa, observamos que a medida que se fue adquiriendo experiencia en el uso de Selenium, la productividad del personal fue creciendo, lo que marca la necesidad de tener testers automatizadores especializados en el área. Un punto importante a destacar en este proyecto en particular, es el alto porcentaje de reutilización de los scripts de una etapa a la otra. De los 271 scripts generados en la segunda etapa, el 40% se basó en scripts generados en la primera etapa, esto contribuyó también al aumento de productividad. Los scripts reutilizados de la etapa 1, pasaron a conformar nuevas suites en la etapa 2, en forma directa o realizándoles pequeños cambios. Cabe aclarar que en este proyecto, la ejecución de dichas suites y los incidentes encontrados por las mismas con cada nueva versión del producto no son parte de las tareas realizadas por el equipo de pruebas del CES, esa tarea la realizaba directamente el cliente.

NÚMEROS DEL PROYECTO		
	Etapa 1 20/11/06 al 18/01/07	Etapa 2 11/03/07 al 03/06/07
Suites	4	26
Scripts	46	271
% Scripts reutilizados etapa anterior	0%	40%
% Scripts nuevos	100%	60%

Tabla 2 – Suites y Scripts generados por etapa

A continuación se describe la experiencia en el uso de la metodología de la sección 2.1 en conjunto con las herramientas de la sección 2.2 en los proyectos de automatización llevados a cabo.

El primer paso en un proyecto de automatización de pruebas funcional es definir aquellas funcionalidades o ciclos funcionales que resulta importante o beneficioso automatizar. También evaluar cuales son los más fáciles de automatizar o los más ejecutados. Esto constituye la actividad AT1 – "Definición de las pruebas automatizadas".

Una vez definidas las funcionalidades, se debe escribir los procedimientos de prueba en los cuales se definen las acciones y verificaciones que conforman las pruebas.

En este paso, es bueno tener en mente las posibilidades que ofrece Selenium en cuanto a las acciones y verificaciones posibles. Selenium permite ejecutar código Javascript arbitrario, por lo tanto, pone a disposición para estos propósitos todo el potencial de este lenguaje.

En el inicio, se comenzó utilizando un documento que especificaba los ciclos funcionales con las funcionalidades que involucra junto con una planilla electrónica donde se enumeran los pasos a seguir en cada pantalla. Luego se notó que era muy costoso llegar a este nivel de detalle y no proporcionaba un beneficio considerable. Se optó por un documento donde se detallan los ciclos funcionales y para cada funcionalidad se especifica las verificaciones que se deben realizar. Los ciclos funcionales y las pruebas de las funcionalidades se transformarán luego en suites y scripts respectivamente. Esto forma parte de la actividad AT2 – "Definición de los procedimientos de prueba".

Una vez definidas las pruebas se puede comenzar a crear y documentar el entorno de datos, esto es AT6 – "Configuración de entorno". Las actividades llevadas a cabo hasta el momento no requieren herramientas más allá de un editor de texto y una planilla electrónica.

La actividad AT3 – "Generación de suites y scripts" comienza cuando están las pruebas definidas y el entorno de datos configurado. En ella los testers encargados de la automatización utilizan Selenium IDE que permite grabar y editar script fácilmente. Esta herramienta, como ya dijimos, es una extensión de Mozilla Firefox. Las extensiones para este navegador se convierten en una herramienta fundamental para el tester automatizador. Si la aplicación no soporta este navegador, los scripts no podrán ser grabados sino que se deberán codificar en forma manual.

Una vez grabados o codificados, los scripts son probados y, en ocasiones, ajustados para que funcionen en otros navegadores. Cabe destacar que el tester requiere aprender sobre la herramienta de automatización Selenium Core, la herramienta de grabación Selenium IDE, el lenguaje HTML-Selenese (de Selenium Core), estándares de programación y procedimientos de gestión de las pruebas definidos por el equipo de automatización. A medida que el desarrollador va adquiriendo experiencia va dejando de lado herramientas como el grabador Selenium IDE y comienza a escribir directamente los script a través de un editor de texto. En nuestro caso utilizamos el entorno de desarrollo Eclipse.
Los scripts se deben mantener bajo gestión de configuración. Para ello se utiliza un servidor CVS donde se colocan todos los artefactos generados en el proceso. Como cliente CVS usamos el propio Eclipse.
Una vez construido el script, se procede a reproducirlo en el servidor Web a través de Selenium Core. Este es un proceso iterativo de ajuste y prueba.
Luego de construidas las suites se procede a ejecutarlas en el orden preestablecido de manera de verificar su correcto funcionamiento y verificar las dependencias. Esto requiere ejecutar secuencialmente en Selenium Core cada una de las suites observando su comportamiento. Esto es parte de AT4 – "Ejecución de las pruebas automatizadas del ciclo funcional".
Luego, se procede con la actividad AT7 – "Validación de las pruebas automatizadas" ejecutando las suites en el ambiente del cliente y verificando el correcto funcionamiento. De esta manera el cliente puede comenzar la validación de las pruebas. Cuando el mismo acepta las pruebas se cierra el proceso.
La actividad AT8 – "Organización de las pruebas automatizadas" se ejecuta en paralelo y durante todo el proyecto. En ella se deben definir, mantener y actualizar los procedimientos de gestión de las pruebas. No se contó con una herramienta que facilite la tarea.
Resulta fundamental la actividad AT5 – "Investigación y modificación de herramientas" pues una de las características más importantes de Selenium es tener a disposición el código fuente. Las posibilidades que ofrece Selenium son amplias (comandos y ejecución de código Javascript del usuario a través del comando *"waitForCondition"*), sin embargo, puede ser necesario realizar extensiones de manera de ajustarnos a características particulares de la aplicación bajo prueba.
Desde que comenzamos a usar Selenium hasta la fecha hemos implementado varias extensiones que nos permitieron nuevas posibilidades en el manejo de ventanas emergentes, soluciones para subir archivos desde el cliente al servidor Web e implementaciones de Selenium Core que ejecutan las pruebas a velocidades diferentes. Los foros de OpenQA son una fuente importante de información.

2.4 Conclusiones de la automatización

De nuestra experiencia práctica en el último año, podemos asegurar la factibilidad de proyectos de automatización de pruebas funcionales para aplicaciones Web utilizando herramientas Open Source.
Los clientes se mostraron satisfechos con el producto resultante de la automatización: scripts, suites y documentación. En los proyectos en lo que hemos participado, los scripts quedaron en propiedad del cliente y son utilizados para ejecutar las pruebas automatizadas sin la necesidad de personal del CES.
En cuanto a Selenium, demostró ser simple, potente y flexible, además de proveer un lenguaje fácil de usar y de aprender. La documentación de Selenium disponible en el sitio se limita a una referencia de comandos, por lo tanto los foros son una importante fuente de información. En general los usuarios de Selenium tienen gran participación en los foros y hay muchas extensiones disponibles de la herramienta.
También debemos destacar algunos aspectos negativos en el uso de Selenium Core, por ejemplo, no se puede obtener datos de fuentes externas como puede ser una base de datos. Sin embargo, si se

utiliza Selenium Remote Control no tenemos este inconveniente, ya que se tiene a disposición todas las posibilidades del lenguaje de programación en el cual se escriben las pruebas. Hay otras dos limitaciones que tiene Selenium Core y que, al igual que en el caso anterior, se pueden sortear utilizando Selenium Remote Control. La primera es que los scripts no manejan instrucciones o comandos de bifurcación, esto es una decisión de diseño de la herramienta, en busca de simplicidad. La segunda es que Selenium Core debe estar instalado en el mismo servidor Web que la aplicación a probar.
Uno de los principales problemas al automatizar aplicaciones utilizando el conjunto de herramientas open source indicadas se encuentra en la administración las suites y scripts resultantes de forma adecuada. Como se muestra en la Tabla 2, en un proyecto de 5 meses de duración se generaron más de 300 scripts, los cuales deben ser organizados y mantenidos. El costo de la administración de los artefactos resultantes de las pruebas no es para nada despreciable. Las tareas relativas a esta área del proyecto se llevaron a cabo manualmente por el equipo de automatización, pero se sintió la necesidad de una herramienta informática que brinde cierta asistencia. Interesa encontrar una herramienta open source que, entre otras cosas, permita administrar los artefactos generados y mantener trazabilidad entre las suites y scripts y las funcionalidades que estas prueban.

3 PROPUESTA PARA LA GESTIÓN DE LAS PRUEBAS AUTOMATIZADAS

A partir de la necesidad de una herramienta que permita gestionar los artefactos generados en la automatización, se investigaron diferentes alternativas para lograr gestionarlos de forma eficiente. Se propone la conjunción de la herramienta FitNesse y Selenium Remote Control para crear y gestionar suites, scripts y documentación de una forma adecuada.
En la sección 3.1 se resume la herramienta FitNesse. En la sección 3.2 se presenta la propuesta para la gestión de las pruebas y en la sección 3.3 se describe el trabajo a futuro para validar dicha propuesta.

3.1 FitNesse

FitNesse es una herramienta para mejorar la comunicación y colaboración en un proyecto de desarrollo de software. Fue creado por Robert Martin, Micah Martin y Michael Feathers, parte del equipo de Object Mentor [19]. El proyecto sigue en desarrollo y utiliza la licencia GNU General Public License versión 2.
FitNesse brinda un espacio donde documentar lo que el sistema debe hacer y compararlo con lo que el sistema hace. En este espacio pueden colaborar cliente, programadores y testers desde el inicio del desarrollo. El cliente puede generar sus propias pruebas, así como ejecutar las pruebas que escribieron los diseñadores de pruebas.
FitNesse es una wiki, lo que aporta facilidad para crear y editar páginas web. Las pruebas se definen mediante tablas dentro de las páginas. Estas pruebas se pueden ejecutar y FitNesse comparará los resultados esperados con los resultados obtenidos.
FitNesse provee interfaces a ser implementadas, mediante los cuales ejecuta una prueba y en caso de ser necesario compara el resultado esperado con el obtenido. Estas interfaces deben ser implementadas en Java.

3.2 Gestión de las Pruebas Automatizadas con FitNesse

Se propone utilizar Selenium Remote Control en Java para implementar las interfaces provistas por FitNesse. Con esto se podrían definir, por ejemplo, los comandos de Selenium Core para luego utilizarlos en los scrips ingresados en las páginas de FitNesse. También crear comandos más sencillos que encapsulen el código JavaScript, logrando así que el diseñador de pruebas se enfoque

en las técnicas de prueba sin preocuparse de aspectos de tecnología y el tester automatizador en diseñar e implementar la interfaz de FitNesse. Esto aumentaría la calidad de las pruebas como también la productividad de la automatización.
Al tratarse de una wiki se podría aprovechar el potencial en cuanto a la gestión de sus páginas. Dentro de las ventajas que ofrece un sistema wiki se encuentran el versionado de las páginas y la asistencia al encontrar inconsistencias en páginas por edición simultánea.

3.3 Trabajo a Futuro

Hasta el momento nuestro trabajo se basó en el uso de la herramienta de automatización Selenium Core y actualmente estamos abordando el trabajo con Selenium Remote Control integrado a FitNesse. A futuro pretendemos implementar suites y scripts más complejos para conocer a fondo las fortalezas y debilidades de estas herramientas integradas.
El objetivo de integrar estas herramientas es crear una interfaz sencilla que permita ejecutar comandos similares a Selenium Core. Para cada aplicación a probar se podría extender la interfaz tomada como base, de manera de definir nuevos comandos y enriquecer el lenguaje. Puede ser útil la colaboración del cliente en la definición del lenguaje aportando comandos que se adapten a sus necesidades.
Otro aspecto interesante que podría surgir en el uso de FitNesse al utilizar Selenium Remote Control con Java es la posibilidad de utilizar todo el potencial que nos provee Java para acceder a bases de datos. De esta forma es posible verificar, por ejemplo, que la aplicación bajo prueba se comporte de la forma esperada consultando para ello la base de datos. Este enfoque no es posible con Selenium Core debido a que permite automatizar pruebas que verifican la aplicación únicamente a través de la interfaz gráfica.
Cabe destacar que el uso de Selenium Remote Control en FitNesse provee una visualización de cómo se van comportando los scripts en su ejecución, característica que no esta presente al utilizar únicamente Selenium Remote Control pues no posee interfaz gráfica para ejecución de las pruebas.

4 CONCLUSIONES

Se presentó una metodología para la automatización de las pruebas funcionales automatizadas y un conjunto de herramientas open source que asisten a dichas actividades.
De nuestra experiencia práctica en el último año, podemos asegurar la factibilidad de proyectos de automatización de pruebas funcionales para aplicaciones Web utilizando este conjunto de herramientas open source. Además, los clientes se mostraron satisfechos con el producto resultante de la automatización: scripts, suites y documentación. Las suites y scripts generados son utilizados para ejecutar las pruebas automatizadas sin la necesidad de personal del CES.
En cuanto a Selenium, demostró ser simple, potente y flexible, además de proveer un lenguaje fácil de usar y de aprender.
Uno de los principales problemas al automatizar aplicaciones utilizando el conjunto de herramientas open source presentadas en este artículo se encuentra en la gestión de los artefactos generados en el proyecto. Como trabajo a futuro, se propone la conjunción de la herramienta Fitnesse y Selenium Remote Control para crear y gestionar suites, scripts y documentación de una forma adecuada.

REFERENCIAS

[1] Apache License 2.0 - http://www.apache.org/licenses/LICENSE-2.0

[2] Beizer B. “Software testing techniques (2nd ed.)”, ISBN:0-442-20672-0, Van Nostrand Reinhold Co, 1990.

[3] Black R. “Managing the Testing Process, 2nd Edition”. ISBN 0-471-22398-0, Editorial Wiley, 2002.

[4] Camara Uruguaya de tecnologías de información (CUTI). http://www.cuti.org.uy/

[5] Centro de Ensayos de Software (CES) .http://www.ces.com.uy, 2007.

[6] Concurrent Version System. http://www.nongnu.org/cvs/

[7] Dustin E., Rasca J., Paul J. “Automated Software Testing”, ISBN 0-201-43287-0, Addison Wesley, 1999.

[8] Eclipse. http://www.eclipse.org/

[9] Firebug - http://www.getfirebug.com

[10] FitNesse - http://FitNesse.org/

[11] Genexus – http://www.genexus.com

[12] IEEE Standard Glossary of Software Engineering Terminology Institute of Electrical and Electronics Engineers,ISBN: 155937067X, 1990.

[13] International Software Testing Qualifications Board, Certified Tester Foundation Level Syllabus, Versión 2005. http://www.istqb.org/fileadmin/media/SyllabusFoundation.pdf

[14] Internet Explorer. http://www.microsoft.com/spain/windows/products/winfamily/ie/default.mspx

[15] Kaner C., Bach J., Pretichord B. “Lessons Learned in Software Testing”, ISBN 0471081124, Wiley, 2001.

[16] Kaner C., Falk J., Nguyen H. “Testing Computer Software, 2nd Edition”, ISBN: 0471358460, Wiley, 1999 .

[17] Mozilla Firefox. http://www.mozilla-europe.org/es/products/firefox/

[18] Myers G. “The art of software testing, 2nd edition”, ISBN 0-471-46912-2, John Wiley & Sons Inc., 2004.

[19] Object Mentor - http://www.objectmentor.com/

[20] OpenQA Sitio Web http://www.openqa.org/

[21] Pérez B., “Proceso de Testing Funcional Independiente (ProTest)”, Tesis de Maestría en Informática, PEDECIBA Informática, Facultad de Ingeniería, Universidad de la Republica, Uruguay, ISSN: 0797-6410 - 06-11, 2006.

[22] Selenium. http://www.openqa.org/selenium/

[23] Selenium Core. http://www.openqa.org/selenium-core/

[24] Selenium Ide. http://www.openqa.org/selenium-ide/

[25] Selenium Remote Control. http://www.openqa.org/selenium-rc/

[26] Software Test Automation, Effective use of test execution tools. Mark Fewster & Dorothy Graham. ISBN 0-201-33140-3

[27] ThoughtWorks. http://www.thoughtworks.com/

[28] Universidad de la República (UdelaR). http://www.universidad.edu.uy/index.php

[29] XPath Checker - https://addons.mozilla.org/en-US/firefox/addon/1095

[30] Xpather - http://xpath.alephzarro.com

BPEL: una propuesta para el uso de Web Services

Patricia Bazán
LINTI. Laboratorio de Investigación en Nuevas Tecnologías Informáticas. Facultad de Informática.
Universidad Nacional de La Plata
Calle 50 y 115 – 1er Piso – la Plata (1900) – Buenos Aires – Argentina
pbaz@ada.info.unlp.edu.ar

Abstract

The notion of service is deeply rooted in the world of business. Service orientation is an organizational principle that is applied to business, governmental environments as well as to software.
A business process is a real-world activity involving a set of logically related tasks that, when performed in the appropriate sequence and according to the correct business rules, produce a business outcome.
Web Services are self-contained, modular business process applications based on industry standard technologies. These standards lack the semantics typical of a business process.
Said semantics may be implemented by means of a language of specification of business processes that imposes restrictions on the services that satisfy them and their behavior.
BPEL is a language of specification of business processes completely executable that allows for Web Services orchestration.

Keywords: Business Process. Service oriented**.** Workflow

Resumen

La noción de servicio tiene profundas raíces en el mundo de los negocios. La orientación a servicios es un principio organizacional que se aplica a los negocios, a los entornos gubernamentales y también al software.
Un proceso de negocio es una actividad del mundo real que consiste en un conjunto de tareas lógicamente relacionadas, que cuando se realizan en la secuencia apropiada y siguiendo las reglas del negocio, producen una salida válida para el negocio.
Los Servicios Web son procesos de negocio modulares y autocontenidos que se basan en tecnologías de estándares de la industria. Estos estándares carecen de la semántica propia del proceso de negocio.
Un lenguaje de especificación de procesos de negocio que imponga restricciones acerca de los servicios que los satisfacen y su comportamiento, es capaz de otorgar dicha semántica.
BPEL es un lenguaje de especificación de procesos de negocio completamente ejecutable que otorga orquestación a los Servicios Web.

Palabras Clave: Proceso de Negocio. Orientación a Servicio. Workflow

1. INTRODUCION

Un proceso de negocio es una actividad del mundo real que consta de un conjunto de tareas lógicamente relacionadas, que cuando se realizan en la secuencia apropiada y siguiendo las reglas del negocio, producen una salida válida para el negocio (realizar un pago, realizar una extracción de efectivo de una cuenta bancaria, etc.)
BPM (Business Process Management) es el nombre de un conjunto de sistemas de software, herramientas y metodologías enfocadas hacia la manera en que las organizaciones identifican, modelizan, desarrollan, distribuyen y gestionan tales procesos de negocios.
Entre las soluciones BMP conocidas se cuentan desde los workflow hasta los servicios de orquestación Web modernos.
SOA (Service Oriented Architecture) basado en Servicios Web, ayudan a alcanzar el objetivo de BPM más rápida y fácilmente.
Desde el punto de vista opuesto, BMP simplifica el problema de cómo combinar la ejecución de múltiples Servicios Web para resolver un caso de negocio particular.
BPM aísla no solamente el proceso respecto de los datos sino que explicita el flujo del mismo y permite separarlo de la forma en que se ejecuta. Esta característica hace que BPM tienda a adoptarse como metodología para conceptualizar problemas más allá de lo tecnológico.
En los orígenes de la ciencia de la computación todo el énfasis estuvo puesto en desarrollar sistemas que automaticen tareas que se hacían manualmente. Este era suficiente desafío.
La tendencia hoy es hacia un paradigma orientado a procesos, donde las aplicaciones deben cubrir la actividad global de la empresa y las herramientas son los BPMS (Business Process Management Sistemas o Sistemas de Gestión de Procesos de Negocio.
Los Servicios Web son procesos de negocio modulares y autocontenidos que se basan en tecnologías de estándares de la industria, pero ninguno de estos estándares les otorga semántica de negocio. [4]
Para otorgar esta semántica es necesario especificar restricciones acerca del uso de las operaciones de de los Servicios Web y su comportamiento, es decir, especificar procesos de negocio.
BPEL (Business Process Execution Language) es un lenguaje de especificación para representar flujos de procesos de una manera adecuada para que una máquina BPEL pueda leerla e interpretarla. Es decir es un lenguaje de especificación completamente ejecutable.
Este trabajo plantea una visión integradora de varios conceptos tecnológicos y no tecnológicos a la luz de construir soluciones que sean capaces de explicitar el conocimiento del problema, sean declarativas y puedan absorber mas rápida y fácilmente los cambios del mundo real.

2. ORIENTACION A SERVICIOS Y LOS PROCESOS DE NEGOCIO

La noción de servicio tiene profundas raíces en el mundo de los negocios. La orientación a servicios es un principio organizacional que se aplica a los negocios, a los entornos gubernamentales y también al software. Para comprenderlo, basta revisar el tipo de servicios que el mundo empresarial y de gobierno presta a clientes, ciudadanos y socios y como los prestan.
Es claro entender el mundo empresarial y de gobierno con la orientación a servicios. Ellos prestan permanentemente servicios como por ejemplo: bancos, agencias de viajes, agencias de seguro, supermercados, oficinas legales, hospitales, transporte, policía, etc.
Las formas de distribución de servicios se pueden clasificar según la siguiente taxonomía [3]:

- **Distribución mediante humanos:** un agente humano es quien se involucra en la distribución del servicio.
- **Auto – Distribución:** el cliente, ciudadano o socio obtiene el servicio usando algún sistema automatizado.
- **Distribución sistema a sistema:** el servicio es automáticamente realizado involucrando generalmente a dos o más sistemas de cómputos.

Cualquier servicio, aun el más simple, puede definirse usando las tres formas de distribución. Se plantea como ejemplo verificar el estado de un vuelo en una línea área.

1- El pasajero llama a la compañía aérea y consulta el estado de su vuelo proveyendo el numero de vuelo (distribución mediante humanos)
2- El pasajero entra a la pagina Web de la línea aérea y busca el estado de su vuelo proveyendo el numero de vuelo (auto-distribución)
3- La línea aérea notifica automáticamente al pasajero del cambio de estado del vuelo vía mail o vía la página Web (según preferencias) y reprograma el alquiler de auto que había realizado el pasajero considerando la demora (distribución sistema a sistema)

El desarrollo orientado a servicios ha sido adoptado por la industria del software como complementario al orientado a objetos, orientado a procesos, orientado a mensajes y orientado a base de datos.
La orientación a servicios a nivel empresarial en los sistemas de información es una tendencia y en ella convergen las siguientes tecnologías claves:

- XML como es un formato de datos independiente y común entre plataformas, lenguajes de programación, ambientes de desarrollo y sistemas operativos.
- Servicios Web como tecnología basada en XML para pasaje de mensajes, descripción de servicios, descubrimiento y características extendidas.
- SOA como una metodología para interoperabilidad y reuso de aplicaciones en IT (Information Tecnology o Tecnología de la Información) con un fuerte foco en los procesos más que en las estructuras de datos.
- BPM como un conjunto metodologías y tecnologías para automatizar los procesos de negocio

Cada una de estas tecnologías tiene su fortaleza propia pero combinadas definen una plataforma completa para obtener los beneficios de la orientación a servicios y dar paso a la evolución de los sistemas IT.
La Tabla 1 plantea los beneficios de la orientación a servicios, describiendo cada uno y ejemplificando el aspecto de aplicación del mismo.

Tabla 1 – Beneficios de la orientación a servicios

Beneficio	Definición	Aplicabilidad
Reusabilidad	Creación de nuevos servicios usando otros	La reusabilidad es una característica que acelera el desarrollo y facilita el mantenimiento.
Eficiencia	Creación de nuevos servicios y nuevas aplicaciones combinando otros	Un servicio se define por los mensajes que intercambia con otros servicios más que por los métodos que firma. [3]

Bajo acoplamiento tecnológico	Capacidad de modelar servicios independientes de su ambiente de ejecución	Los servicios poseen un nivel de abstracción superior que el de un objeto ya que es posible mapearlo a un lenguaje orientado a procesos (COBOL, PL/1) o a un sistema de cola de mensajes (JMS o MSMQ) o a un sistema orientado a objetos (J2EE o .NET).
División de responsabilidades	La gente de negocios se concentra en los aspectos del negocio y que la gente técnica se concentra en los aspectos tecnológicos	La creación del servicio trata con la complejidad de la tecnología subyacente sobre la que servicio estará implementado. El consumo del servicio ensambla nuevas aplicaciones y flujos de procesos de negocio, asegurando que los datos y flujos reflejen los requerimientos operacionales del negocio.

3. CONCEPTOS DE SOA

SOA es un estilo de diseño que guía todos los aspectos de crear y usar servicios de negocio a través de su ciclo de vida, definiendo y proveyendo una infraestructura de IT que permita que diferentes aplicaciones intercambien datos y participen en procesos de negocios en forma independientes de lenguajes, sistemas operativos y sistemas de red subyacentes.

Los grandes objetivos de SOA tienen que ver con:

- Ayudar a alinear las capacidades de la IT con los objetivos del negocio
- Proveer una infraestructura técnicamente ágil para responder fácil y rápidamente a los cambios de requerimientos.

El éxito de SOA radica en determinar el diseño y funcionalidad correcta de los servicios dentro de una librería de servicios reusables que puedan ser combinados y alineados para responder a las características operacionales del negocio.

El concepto de separar la interfase de su implementación ya fue introducido por las tecnologías J2EE, CORBA y COM. El aporte de SOA es lograr una separación limpia y completa separando la descripción del servicio de su implementación, básicamente interpretando un archivo XML y esta capacidad es parte de los conceptos y tecnologías que aportan los Servicios Web.

Además, un servicio es un elemento de alta disponibilidad y otorga un grado de especialización en la solución de los problemas que contribuye a manejar mejor la complejidad.

Retomando el concepto de división de responsabilidades planteados en el apartado anterior, existen dos perspectivas desde donde comprender el concepto de servicio:

- Desde la perspectiva de negocio, los servicios son activos de la IT que corresponden a actividades de negocio del mundo real o funciones de negocio reconocidas que pueden accederse de acuerdo a políticas establecidas.
- Desde la perspectiva técnica, los servicios son unidades de granularidad gruesa, activos de la IT que tienen interfases bien definidas y separadas de la implementación.

Este concepto en SOA se denomina “abstracciones a nivel de servicio” porque es capaz de tratar exclusivamente con los servicios y además definir todos los elementos del mismo.
La figura 1, muestra gráficamente lo enunciado [3].

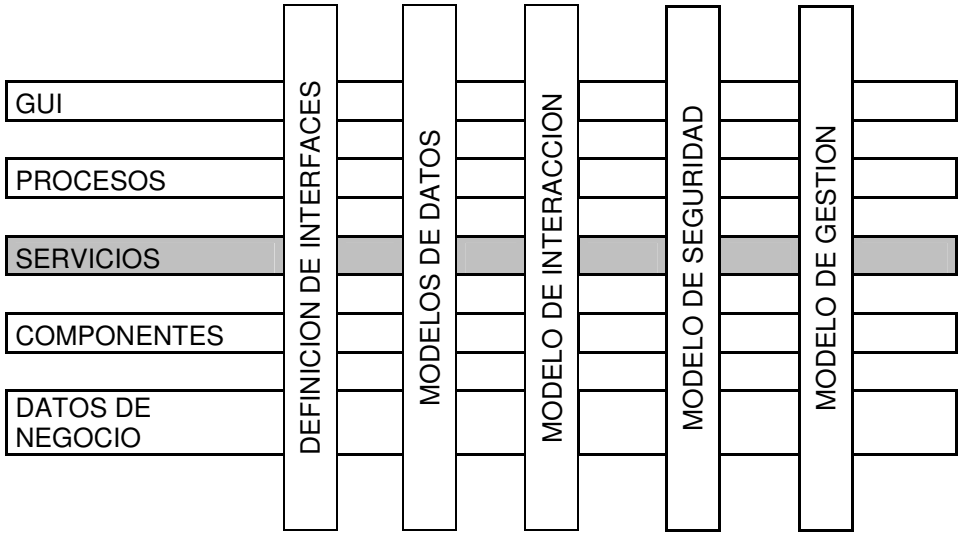


Figura 1. Abstracciones a nivel de servicio

En esta figura se pone en contexto a los servicios y su nivel de abstracción. Las filas representan los niveles existentes en cualquier arquitectura empresarial. Las columnas son los cortes transversales que se aplican a los distintos niveles.
SOA trata exclusivamente con el nivel de servicio. Por ello, si bien el modelo se seguridad atraviesa todos los niveles y trata con la seguridad en todos sus aspectos (desde el acceso al medio físico hasta el control de acceso, pasando por el manejo de palabras claves) está claro que SOA trata solamente con los aspectos de seguridad vinculados al servicio.

4. CONCEPTOS DE BPM

Un proceso de negocio es una actividad del mundo real que consiste de un conjunto de tareas lógicamente relacionadas que si se realizan en la secuencia apropiada y de acuerdo a las reglas de negocio correctas, producen una salida de negocio.
BPM determina la manera en que una organización puede identificar, modelar, desarrollar, distribuir y administrar sus procesos de negocio, incluyendo aquellos procesos que involucran sistemas de IT e interacción humana.
BPM tiene sus raíces en el workflow y ha progresado sobre los sistemas de orquestación de WS (Web Services o Servicios Web) actuales.
Los objetivos y beneficios de BPM son:

- Reduce las errores de obstrucción entre requerimientos de negocio y los sistemas de IT ya que los usuarios de negocio modelizan los procesos y luego el departamento de IT provee la infraestructura para ejecutarlos.
- Incrementa la productividad de los empleados

- Incrementa la flexibilidad y agilidad corporativa separando la lógica del proceso de otras reglas de negocio. Esto absorbe mejor los cambios en los requerimientos.
- Reduce los costos de desarrollo con lenguajes de programación grafica de alto nivel.

La automatización de los procesos de negocio incluye la automatización y rastreo de procesos de negocio, durante el cual los documentos y/o tareas circulan entre los participantes para accionar según un conjunto de reglas preestablecidas.

Los sistemas de IT implementan procesos de negocio, sin embargo BPM, al separar la lógica de negocio de la reglas, esta explicitando las mismas.

4.1. BPM Systems

BPM es la disciplina para definir, administrar y ejecutar procesos de negocio como un activo corporativo.

BPMS proveen la tecnología que implementa uno o más de estas funciones centrales.

Muchos BPMS proveen una herramienta de modelizacion que permite que los procesos se definan como un grafo donde los nodos representan la tarea y los arcos el flujo de control y dependencias de tareas.

Un BPMS debe proveer:

- **Modelizacion de procesos**: permite capturar los requerimientos de negocio en su etapa inicial y ponerla disponible durante el resto del proceso de desarrollo
- **Ejecución de procesos:** la maquina de ejecución de procesos de la suite BPM importa el proceso modelado (definido usando BPEL) y luego ejecuta y administra las instancias de procesos para alcanzar los requerimientos operacionales.
- **Monitoreo de procesos:** esta capacidad incluye ver el resumen de los procesos en ejecución, de los completados, ver estados de procesos, suspender y reanudar procesos, dar alertas y reasignar procesos.
- **Monitorear la actividad de negocio:** (BAM Business Activity Monitoring) analiza los eventos generados por la actividad de negocio y permite dar metricas.

Planteadas estas características esta claro que si bien WS-BPEL es vital en la pila de WS, no puede verse como una solución BPM completa para WS. WS-BPEL es un lenguaje para representar flujos de procesos de una manera adecuada para que una maquina WS-BPEL pueda leerla e interpretarla. WS-BPEL no pretende direccionar o estandarizar ningún otro aspecto de una solución BPM.

5. CONCEPTOS DE SERVICIOS WEB

Los Servicios Web son una nueva generación de aplicaciones Web. Son componentes de software autocontenidas, autodescriptivas y modulares que pueden ser accedidas, localizadas e invocadas desde cualquier lugar sobre la Internet [1]. Dan soporte a la prestación de servicios entre sí y también al descubrimiento de dichos servicios Se construyen sobre estándares como UDDI, WSDL y SOAP.

Los Web Services son:

- Publicados y localizados vía UDDI (Universal Description, Discovery and Integration).
- Descriptos usando WSDL (Web Service Description Language).
- Invocados vía SOAP (Simple Object Access Protocol) sobre HTTP.

Por último, con XML (extensible markup language) como lenguaje de descripción de datos se obtiene la interoperabilidad necesaria ya que el mismo es independiente de las tecnologías, las plataformas y la representación física de los datos.

Un SOA basado en Web Services tiene la siguientes ventajas:

- Está basado en estándar (no propietario)
- Provee interoperabilidad de soluciones
- Soporta integración intra-organización, a través de la organización y entre organizaciones.

5.1. Plataforma Web Services

La plataforma WS esta basada en estándares que son neutrales a productos, tecnologías y middleware que pueden integrar servicios usando una variada gama de productos, plataformas, tecnologías y middleware.

Los elementos claves que componen la plataforma son:

- **Contrato de servicio**: interfase no ambigua y bien definida en WSDL.
- **Repositorio de contratos de servicio**: una base de datos para almacenar, buscar y versionar servicios.
- **Registracion y búsqueda (lookup)**: es un servicio de nombres para ubicar "instancias " de servicio y recursos en ejecución. Se diferencia del punto anterior es que aquel es estático y este dinámico.
- **Nivel de servicio de seguridad**: incluye autenticación, control de acceso, contexto de autorización, firma, privacidad, integridad y no-repudio. (no se trata de seguridad a nivel de aplicaciones sino de servicios)
- **Nivel de servicio de gestión de datos**: repositorio XML para almacenar y administrar representaciones de datos a nivel de negocio. Si se construye independiente del contrato de servicio, este repositorio puede incluirse en documentos XML usados para validación de datos, transformación, maquina de reglas, etc.
- **Nivel de servicio de comunicación:** soporta múltiples patrones de comunicación e interacción usando SOAP.
- **Nivel de servicio de calidad:** alta confiabilidad, ordenamiento de mensajes, garantía de distribución, manejo transaccional, alta disponibilidad (clustering, auto recuperación, balanceo de carga, hot-deployment)
- **Nivel de servicio de gestión:** soporte de distribución, inicio, detención y monitoreo de servicio.

Entre los principios fundamentales de un WS existen dos puntos contrapuestos: por un lado se define al servicio como un elemento mínimo, pequeño y simple y por otro se le pide absorber gran parte de la complejidad de las aplicaciones (single sign.-on, control de acceso basado en roles, log de auditoria, mensajes confiables y manejo transaccional)

Este es el verdadero compromiso de construcción de la plataforma.

6. BPM, SOA Y WEB SERVICES: VISIÓN COMBINADA

Muchas organizaciones tienen varias capas de aplicaciones y tecnología casi todas de naturaleza stand-alone, donde es difícil compartir información por diferencias de plataformas, modelos de datos, etc.

SOA y WS introduce el concepto de nivel de servicio que consiste en proveer una capa más de servicios de negocio alineados a un dominio de negocio. Estos servicios pueden compartirse transversalmente entre dominios. La plataforma de WS permite definir y utilizar estos servicios independientemente de aplicaciones y tecnología.

El nivel de servicio provee:

- funcionalidad de alta granularidad que mapea tareas de negocio en procesos de negocio
- los contratos de servicio otorgan una interfase bien definida y no ambigua para acceder a los servicios

- el registro y descubrimiento de servicios asegura que el nivel de proceso de negocio pueda ubicar dinámicamente el servicio necesario
- el nivel de servicio de modelo de datos se define sobre el dominio del negocio independiente de modelo subyacente. XML es el formato canónico para intercambio de información.
- la seguridad provee single sign-on y asegura el control de acceso

BPM sin nivel de servicio es complejo y frágil. Complejo porque se debe acceder directamente a las aplicaciones existentes usando interfases ad-hoc para cada caso. Es frágil porque al estar tan ligado a las aplicaciones, cualquier cambio en ellas impacta en la interfase.

7. GESTIÓN DE PROCESOS DE NEGOCIO Y WEB SERVICES: WS-BPEL

Los WS son procesos de negocio modulares y autocontenidos que se basan en tecnologías de estándares de la industria: WSDL para describir, UDDI para darse a conocer y SOAP para comunicarse. Sin embargo, ninguno de estos estándares otorga semántica de negocio a los WS [4]
Para otorgar esta semántica falta especificar restricciones acerca del uso de las operaciones de la colección de WS y su comportamiento. Esto equivale a especificar procesos de negocio.
BPEL para WS permite especificar como los procesos de negocio se relacionan con los WS.
Los procesos de negocio especificados con BPEL son completamente ejecutables y portables.
Un proceso de negocio BPEL interopera con los WS de sus pares, aun si los WS no están implementados con BPEL.
Un proceso BPEL se expone como servicio definido con WSDL y es invocado por otro WS.

7.1. Definiciones

Los procesos de negocio especificados vía BPEL prescriben el intercambio de mensajes entre WS. Estos mensajes son mensajes WSDL. Algunos de estos mensajes pueden incluirse en lo que se denomina "contexto de negocio" de los procesos de negocio. Este contexto es un conjunto de mensajes WSDL llamado "contenedor" y representa los datos que son importantes para la correcta ejecución del proceso de negocio, por ejemplo decisiones de ruteo. Es habitual que este contexto de negocio requiera persistencia.
La idea de "contenedor" se asocia a cada tarea en el flujo, cada una tiene un esquema de definición. En correspondencia a un "contenedor" hay un mensaje. Un "contenedor" es un WS con información adicional acerca de cómo procesar el mensaje e indica cual es el próximo paso a dar.
Entre las tareas básicas para crear composiciones de WS se encuentran:

- Invocar tareas por parte de los procesos de negocio y mediante modelos de interacción one-way o request/reply.
- Recibir tareas bloqueando las mismas en espera de arribos.
- Responder tareas mediante el envío de mensajes de respuesta por parte de los los procesos de negocio.
- Esperar tareas estableciendo tiempos de espera para los procesos.
- Asignar tareas mediante la copia datos desde un lugar a otro
- Lanzar tareas en forma asincrónica, por ejemplo, cuando ocurre un error
- Terminar tareas de una instancia de orquestación entera. Las tareas estructuradas se usan para combinar las anteriores en procesos complejos
- Secuenciar tarea, dando un orden de ejecución.
- Intercambiar tareas seleccionando una rama de la lógica condicional
- Bloquear y esperar por un mensaje

- Definir un grupo de tareas que se ejecuten repetidamente, estableciendo lazos repetitivos.
- Establecer un conjunto de pasos a ejecutar en paralelo

Las "actividades" en BPEL son las actividades que se llevan a cabo dentro del proceso de negocio.
Los "manejadores de falla" permiten atrapar y tratar errores. BPEL provee mecanismos de recuperación de errores.
Un "alcance" es una actividad estructurada que permite agrupar actividades. Además permite definir un contexto de ejecución común para sus conjuntos de actividades.
Las acciones que requieren deshacer actividades completas se denominan "manejadores de compensación".
BPEL, a través del mecanismo de alcance define un conjunto de actividades que deben ser deshechas ante una situación de error. Tales conjuntos de actividades so una unidad lógica de trabajo o transacción. La diferencia con las transacciones de una BD es que estas son cortas y establecen cerramientos ("locking") de los recursos. Las transacciones en BPEL pueden ser mas largas, no se establecen cerramientos y se utilizan las acciones de compensación para cumplir con las propiedades ACID (Atomic, Consistency, Isolation, Durability)
Las aplicaciones creadas con BPEL están basadas en procesos. Se divide la aplicación en dos niveles claramente separados: el nivel superior de los procesos de negocio escritos en BPEL y que representan el flujo de la lógica de aplicación, mientras que a nivel inferior, los WS representan la funcionalidad lógica de la aplicación.
Durante su tiempo de vida, la instancia del proceso de negocio mantiene conversaciones con su "partener". En este caso se requiere un mecanismo a nivel de aplicación que aparee mensajes con conversaciones. BPEL dirige los escenarios de correlación dando un mecanismo declarativo para especificar conjuntos de correlación.
Se puede tener dos visiones de BPEL: por un lado puede verse como un script XML que una maquina de procesos de negocio ejecuta. Pero por otro lado puede verse como un lenguaje de intercambio, o sea la maquina de procesos de negocio permite convertir un lenguaje propietario a BPEL y viceversa.

7.2. Orquestación y coreografía

Los conceptos de orquestación y coreografía son usados para describir dos enfoques que permiten componer los Servicios Web.
La orquestación impone el orden y el ritmo individual a un conjunto de WS del mismo modo que un director de orquesta ordena a los músicos.
La coreografía define el comportamiento variado entre los WS, de la misma manera que un coreógrafo dirige a los bailarines en un baile coreográfico para que realicen su tarea en un espacio e interactuando entre si.
BPEL se enfoca sobre la orquestación y la composición de servicios individuales y puede usarse para dirigir la interacción a través límites empresariales.
Se puede decir que BPEL "orquesta" los Servicios Web.

8. CONCLUSIONES

En el presente trabajo se puso énfasis en obtener una idea conceptual de distintas tecnologías y metodologías con el fin de obtener una visión convergente de las mismas.
Por un lado, esta claro que la industria del software esta dando un giro hacia las metodologías más formales que capturen mejor el conocimiento del negocio y que además sean fáciles de implementar, más declarativas y fundamentalmente adaptables a los cambios.

Desde lo tecnológico el concepto de servicio como elemento siempre disponible, autocontenido y con capacidad de ejecución es también una nueva tendencia.
La convergencia de estos enfoques es exitosa si contamos con un elemento que aglutine los conceptos enunciados y que además permita construir soluciones que puedan expresarse, ejecutarse y modificarse en el mismo lenguaje y con la misma metodología.
BPEL surge como una propuesta de uso de Servicios Web y además como el lenguaje de especificación de procesos de negocio netamente declarativo con capacidad de absorber conocimiento implícito, explicitándolo.
Se puede tener dos visiones de BPEL: por un lado puede verse como un script XML que una maquina de procesos de negocio ejecuta. Pero por otro lado puede verse como un lenguaje de intercambio, o sea la maquina de procesos de negocio permite convertir un lenguaje propietario a BPEL y viceversa.
Es de interés continuar con trabajos futuros que completen las visiones posibles de BPEL y elaboren una evaluación de posibles herramientas que plasmen los conceptos aquí enunciados.

9. REFERENCIAS

[1] Ben van Eyle. Web Services – A Business Perspective on Platform Choice. *www.theserverside.com*. 2001.

[2] Bazán-Rodriguez. Java y .NET Comparación de Paradigmas. *5ta Conferencia Iberoamericana en Sistemas, Cibernética e Informática. CISCI 2006. Orlando, Florida*. Julio 2006

[3] Eric NewComer – Grez Lomow. Understanding SOA with Web Services. *Addison-Wesley.* 2005

[4] Frank Leymann, Dieter Roller. Bussiness Processes in a Web services world. 2002

[5] Shyman Kumar Doddavula, Sandeep Karamongikar. Designing an Enterprise Application Framework for Service-Oriented Architecture.
http://today.java.net/pub/a/today/2005/04/28/soadesign.html. 2005

[6] Margie Virdell. *Business Porcesses and workflow in de Web services world*. 2003

Combinação e Aplicação de Técnicas para o Desenvolvimento de Software Orientado a Aspectos

Gabriel Costa Silva, Munif Gebara Junior, Daniela Eloise Flor
Curso de Sistemas de Informação, Universidade Paranaense (UNIPAR)
Paranavaí - PR, CEP 87706-490, Brasil
munif@unipar.br

e

Yandre Maldonado e Gomes da Costa
Departamento de Informática, Universidade Estadual de Maringá (UEM)
Maringá - PR, CEP 87020-900, Brasil
yandre.costa@gmail.com

Abstract

This work describes obtained results with the combination of two techniques for aspect oriented software development. A method to aspects and components development was applied into requisite's document of system, producing documents for a software development. A modeling technique for aspects was combined with method in one of his phases, producing new documents. Both this techniques combination, allows this aspects is detected and modelled during the development process in the modular form, allowing the reuse and considering both components and aspects as "first-class citizens".

Keywords: Software Development Method, Aspect-Oriented Development, Components, Crosscutting Concern, Software Modelling.

Resumo

Este trabalho descreve os resultados obtidos com a combinação de duas técnicas para o desenvolvimento de software orientado a aspectos. Um método para o desenvolvimento baseado em componentes e aspectos foi aplicado sobre o documento de requisitos de um sistema, gerando artefatos para a produção de um software. Uma técnica de modelagem própria para aspectos foi combinada ao método em uma de suas fases, produzindo novos artefatos. A combinação destas técnicas permite que aspectos sejam detectados e modelados durante o processo de desenvolvimento de forma modular, permitindo o reuso e tratando tanto os componentes-base quanto os aspectos como "cidadãos de primeira classe".

Palavras chaves: Método de Desenvolvimento de Software, Desenvolvimento Orientado a Aspectos, Componentes, Interesses Transversais, Modelagem de Software.

1 INTRODUÇÃO

Com o constante aumento da competitividade no mercado mundial, os sistemas computacionais se tornaram indispensáveis nas empresas. Através deles é possível alcançar uma maior produtividade com menor custo. Alguns sistemas permitem que simulações sejam feitas, evitando que erros sejam cometidos e garantindo uma maior qualidade no produto ou serviço oferecido.
Com o aumento da demanda por *software*, a comunidade de desenvolvimento tem buscado meios de garantir uma maior produtividade aliada ao aumento da qualidade no produto computacional. Essa busca por qualidade fez surgir metodologias de desenvolvimento que contemplam o software desde os interesses do usuário até sua implantação na empresa. Técnicas de modelagem, documentação e modularização surgiram no intuito de permitir que softwares sejam alterados de forma simples e rápida [12].
Este trabalho aborda o desenvolvimento de *software* baseado em componentes e aspectos. O uso de componentes torna o desenvolvimento menos complexo, mais ágil e oferece maior qualidade para o produto computacional [5]. A utilização de uma metodologia de desenvolvimento garante que todos os ciclos de vida do *software* serão contemplados, permitindo que ajustes sejam feitos ainda em nível de projeto, diminuindo assim a probabilidade de erros futuros. Uma linguagem de modelagem auxilia na comunicação entre os envolvidos no projeto do sistema, fornecendo notações comuns para todas as áreas envolvidas [1]. Por fim, a separação de interesses promovida pela utilização da orientação a aspectos, permite que o software se torne mais modular a medida que o código referente ao domínio da aplicação é separado do código que implementa outros interesses, como os interesses não-funcionais.
Para a elaboração deste trabalho, um experimento foi conduzido tomando-se por base o documento de requisitos de um sistema para prestadores de serviço. O método para desenvolvimento de *software* orientado a aspectos (DSBC/A), proposto por [5], foi aplicado sobre este documento, gerando artefatos. A linguagem aSideML foi introduzida na etapa de Especificação sobre os modelos que envolvem aspectos, e sobre os modelos que envolvem o relacionamento de aspectos e elementos base. Na etapa de Provisionamento e Montagem, alguns componentes foram providos através de repositórios de componentes, e aqueles não encontrados, foram implementados. Todos os aspectos foram implementados utilizando a linguagem AspectJ.

2 PROGRAMAÇÃO ORIENTADA A ASPECTOS

Para o desenvolvimento de um produto de *software* é necessário que os interesses do usuário sejam decompostos em unidades menores de *software*. Essas unidades colaboram entre si para que um determinado objetivo do sistema seja atingido. Esse princípio recebe o nome de modularização. No entanto, existem interesses que não podem ser modularizados em uma única unidade. Esses interesses atravessam outras unidades, gerando o espalhamento (*scattering*) e entrelaçamento (*tangling*) de código [9]
Para exemplificar, vamos supor o controle de *logging* de um sistema. O controle de *logging* permite armazenar em registros as atividades realizadas pelo sistema como o acesso de usuários, tentativas de invasão e falhas do sistema, por exemplo. Neste caso, a parte do código fonte referente ao domínio do negócio visa permitir que um usuário tenha acesso ao sistema, ou que um recurso seja corretamente compartilhado. Entretanto, para que haja o controle dessa atividade, é necessário que o código seja atravessado por um segundo código, não inerente ao negócio, mas que tem por objetivo o monitoramento dessa atividade. Esse entrelaçamento de interesses é chamado de interesse transversal (*Crosscutting concerns*) [3], e fez surgir a necessidade de formas de se separar código referente ao negócio, dos interesses não inerentes ao negócio de uma forma centralizada [11]. Padrões de desenvolvimento (*design patterns*) são utilizados em alguns casos com o intuito de

diminuir o problema dos interesses transversais. No entanto, mesmo com o uso de padrões, a redundância ainda não é eliminada por completo.
Através do uso da orientação a aspectos, é possível modularizar os interesses transversais em unidades centralizadas, que afetam os componentes desenvolvidos para suprir os interesses de negócio inserindo neles comportamentos ou mesmo atributos, sem que o código referente ao negócio seja entrecortado [9].
Na orientação a aspectos, os interesses de negócio são implementados através de uma linguagem base, como Java, por exemplo, enquanto os interesses não referentes ao negócio, são implementados como aspectos, utilizando-se uma linguagem própria, como AspectJ, por exemplo. Dessa forma, a orientação a aspectos afeta as unidades base, mas estes, não precisam conhecer os aspectos.
A programação orientada a aspectos é composta de uma linguagem para implementar os interesses base, que será utilizada para desenvolver o código referente ao negócio da aplicação. É necessária ainda uma linguagem para o desenvolvimento de aspectos. Com o programa base já escrito, o código referente a aspectos é incorporado ao código base através de um combinador de aspectos (*aspect weaver*), inserindo os novos comportamentos ou atributos, e gerando uma nova aplicação. Este esquema é apresentado com detalhes na figura 1.

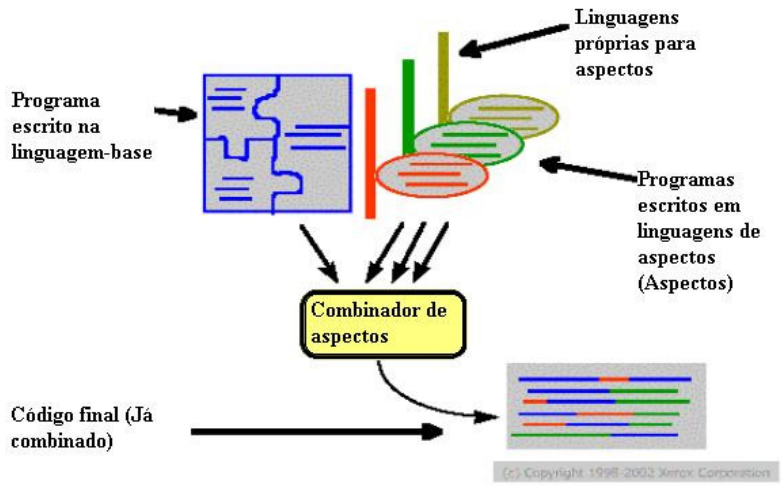


Figura 1: Processo de combinação de aspectos e linguagem-base [9].

3 DESENVOLVIMENTO BASEADO EM COMPONENTES

A modularização está inserida no conceito de engenharia de *software* e faz parte da evolução das linguagens de programação [5]. A diminuição da complexidade do desenvolvimento é um dos fatos que justificam a importância do uso da modularização na elaboração de *software*. Pode-se mencionar um exemplo bastante emblemático relacionado à indústria automobilística. Imagine como seria complexa a criação de um carro no qual todas as peças fizessem parte de uma única estrutura, sem separações. Caso uma das peças apresentasse problemas, toda sua estrutura deveria ser refeita, gerando enormes perdas de tempo e dinheiro. A modularização permitiria a substituição de uma peça sem que as demais sejam afetadas por essa mudança.
Da mesma forma, uma peça, como um volante por exemplo, pode ser usada em vários veículos de diferentes modelos, oferecendo sempre a mesma funcionalidade. Este princípio recebe o nome de reuso e é aplicado no desenvolvimento de *software*. O reuso de uma unidade de *software* já existente, diminui o esforço durante o desenvolvimento, visto que menos unidades deverão ser implementadas. Outra vantagem advinda do reuso é o aumento da qualidade, uma vez que uma unidade que já foi reutilizada algumas vezes, obrigatoriamente já passou pelo teste de outros

desenvolvedores, o que atesta a favor da sua qualidade [8]. Além das vantagens, outro ponto que motiva o reuso de *software* é a diminuição do tempo de desenvolvimento, pois, atualmente, não é mais viável criar um *software* a partir do zero [5].
A idéia de reuso gerou diferentes métodos de desenvolvimento, o desenvolvimento baseado em componentes (DBC) é um deles. No DBC, a principal idéia é a construção de *software* a partir de algo já existente [8]. Um componente encapsula serviços, que são oferecidos através de uma interface bem definida [5]. O uso de interfaces para acessar um componente permite que esse componente seja atualizado ou mesmo substituído sem a necessidade se alterar os demais componentes. No DBC, toda comunicação é feita através das interfaces, sendo que os componentes possuem interfaces oferecidas e requeridas [8]. A figura 2 mostra dois componentes e suas respectivas interfaces oferecidas e requeridas. O componente Credito oferece uma interface para acesso aos seus serviços, chamada IFornece. Por sua vez, esse componente depende de uma interface – IConsulta, fornecida pelo componente Cliente. Nesse exemplo, o componente Cliente poderia passar por alterações, ou mesmo ser substituído por uma versão diferente, sem gerar problemas para o componente Credito, pois o acesso está sendo feito por suas interfaces.

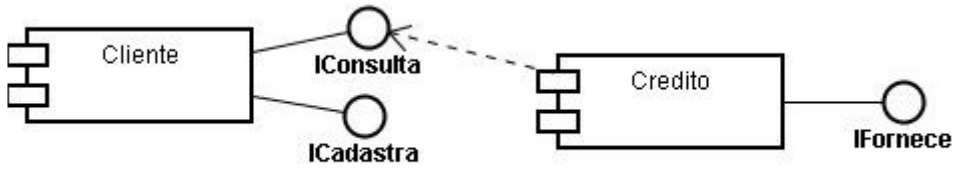


Figura 2: Exemplo de componentes e interfaces oferecidas e requeridas.

Outro ponto a ser considerado sobre componentes são os detalhes de implementação. Um componente deve ser visto como uma "caixa preta" para o usuário (desenvolvedor). Isso acontece porque, ao desenvolvedor da aplicação, cabe conhecer os serviços oferecidos pelo componente, e não a forma como esses serviços são implementados internamente. É importante destacar que componentes são unidades independentes. A união desses componentes, através dos relacionamentos de oferta/dependência de suas interfaces, gera um sistema completo e funcional [5].

3.1 Método Para Desenvolvimento de *Software* Baseado em Componentes e Aspectos

O objetivo do método para desenvolvimento de *software* baseado em componentes e aspectos (DSBC/A) proposto por [5] é, partindo do documento de requisitos do sistema, produzir uma arquitetura formada por componentes base e aspectos. O método DSBC/A opera sobre as interfaces, preservando o encapsulamento dos componentes [5].
A figura 3 apresenta as etapas do método DSBC/A, os principais artefatos gerados e utilizados em cada etapa segundo apresentado por [5]. Cada etapa é subdividida em etapas menores, formadas por algumas atividades. O documento de requisitos é o artefato inicial, utilizado para alimentar a primeira atividade na etapa de Análise de Requisitos. Durante esta etapa, os requisitos do usuário são processados por diversas atividades. Entre as atividades desta etapa, está a identificação dos primeiros requisitos candidatos à implementação com aspectos. Na etapa de Especificação são produzidas as especificações para os componentes base e transversais, bem como as interfaces requeridas e oferecidas. Esta etapa é importante, pois entre outras atividades, é nela que os primeiros candidatos a aspectos são expostos a critérios para a identificação dos componentes transversais. Nas etapas seguintes os componentes que satisfazem os requisitos do sistema e do projeto realizado são providos e por fim, unidos. A interface com o usuário é então implementada

produzindo uma aplicação completa [5]. As etapas de Teste e Implantação não são abordadas por [5] em seu trabalho.

4 MODELAGEM DE *SOFTWARE*

Atualmente, a *Unified Modeling Language* (UML) é um padrão amplamente aceito e empregado para a modelagem de *software* na indústria. Baseada na orientação a objetos (OO), UML é uma linguagem de propósitos gerais, que oferece visões para a descrição de um sistema, desde requisitos até a implementação. Seus diagramas oferecem visões tanto para o comportamento estático, quanto para o comportamento dinâmico do sistema, além de ser uma linguagem independente de processo, ferramenta ou linguagem de programação. A UML ainda apresenta uma forte característica que é a sua extensibilidade. Os mecanismos de extensão da UML permitem que ela seja utilizada para representar novos modelos, que possam surgir com a evolução da complexidade dos sistemas [3].

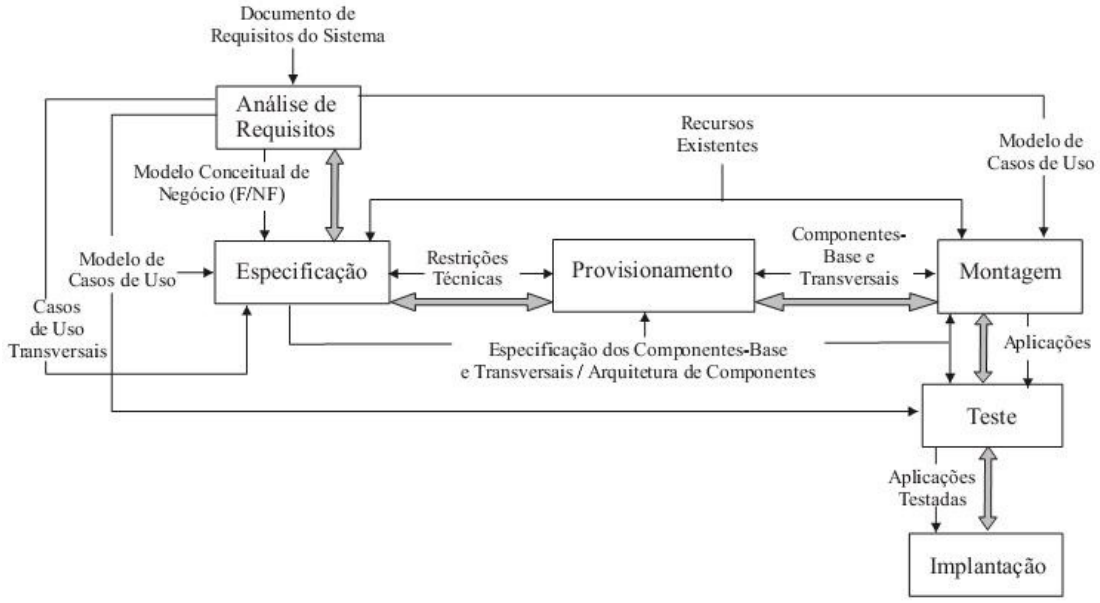


Figura 3: Etapas do método para DSBC/A [1].

Na programação orientada a aspectos (POA), os interesses transversais devem coexistir com os blocos básicos convencionais, independentemente destes serem ou não orientados a objeto. O aspecto deve ser tratado como um bloco adicional, relacionado por um relacionamento de *crosscutting*. Um relacionamento de *crosscutting* especifica a dependência entre um aspecto e um elemento base [3]. A modelagem orientada a aspectos (MOA), incorpora esses conceitos. Segundo [3], "A MOA propõe o uso de outros elementos que nomeiam e localizam interesses transversais e o uso de outras visões a fim de melhorar a compreensão, além de facilitar a reutilização e a evolução". Uma linguagem para MOA deve oferecer compatibilidade com a UML, além de possuir notação independente de linguagem de programação [3].

Existem algumas abordagens que avaliam o uso da UML para a orientação a aspectos, como a UMLaut , AODM e a *Composition Patterns*. De forma geral, essas abordagens geram a necessidade de extensão, os interesses transversais são nomeados, mas não explicitamente especificados, além disso, falta suporte para expressar a semântica de *crosscutting* [3]. Segundo [3], "aspectos não são classes, portanto, a abstração de aspectos deve ser especificada por um novo elemento de modelagem".

4.1 Linguaguem aSideML

A aSideML é uma linguagem baseada na UML que oferece semântica, notação e regras para modelos orientados a aspectos. A aSideML fornece visão para os modelos estáticos e dinâmicos do sistema, além dos modelos composicionais. Os modelos composicionais apresentam o processo de combinação, que envolve a combinação dos elementos base do sistema com os elementos transversais (aspectos). Na aSideML, aspectos e *crosscutting* são tratados como "cidadãos de primeira classe" [3]. Conforme pode ser observado na tabela 1, os elementos de modelagem oferecidos pela aSideML podem ser estruturais ou comportamentais. Os principais elementos estruturais são: o aspecto, elementos base e os relacionamentos. Os principais elementos comportamentais são instâncias de aspecto, interações aspectuais e colaborações aspectuais [3]. A aSideML fornece novos diagramas e enriquece alguns da UML tradicional.

Tabela 1: Modelos, diagramas e elementos da aSideML [4].

Modelo	Diagramas	Perspectivas	Elementos
estrutural	diagrama de aspectos		aspecto, interface transversal, característica transversal
	diagrama de classe estendido	centrado em aspecto	aspecto, *crosscutting*
		centrado na base	interface transversal, *order*
comportamental	diagrama de sequência estendido	ponto de combinação	ponto de combinação dinâmico
	diagrama de colaboração aspectual		instância de aspecto, colaboração aspectual
	diagrama de sequência		instância de aspecto, interação aspectual
processo de combinação	diagrama de classes combinadas		classe combinada
	diagrama de colaboração combinada		colaboração combinada
	diagrama de sequência combinada		interação combinada

5 COMBINAÇÃO E APLICAÇÃO DE TÉCNICAS

Esta seção apresenta os resultados obtidos com a combinação do método para DSBC/A, proposto por [5], e a linguagem de modelagem aSideML, proposta por [3]. O método para DSBC/A é formado por um conjunto de atividades que permitem o desenvolvimento de componentes e aspectos desde o documento de requisitos até a interface com o usuário. Já a linguagem de modelagem aSideML, fornece notações e regras que permitem a construção de modelos orientados a aspectos [3].

O método para DSBC/A é um método completo, no entanto, a modelagem de aspectos e a modelagem que envolve aspectos e componentes base é feita utilizando extensões da UML. Estas extensões não permitem que sejam visualizadas todas as características de aspectos e, em alguns momentos, torna difícil a leitura de alguns modelos. A linguagem aSideML foi projetada para ser independente de linguagem e metodologia de desenvolvimento [3].

A combinação de ambas as técnicas permitem um desenvolvimento orientado a aspectos, onde os aspectos, componentes base e seus relacionamentos, são identificados, modelados e implementados considerando todos suas características, independente da linguagem de programação utilizada.
As etapas de Análise de Requisitos e Especificação são as etapas onde são elaborados os artefatos que definem a arquitetura do sistema. Por este motivo, estas etapas serão abordadas com ênfase neste trabalho. É na etapa de Especificação que a linguagem aSideML é introduzida. Nesta etapa são definidos os elementos base do sistema, bem como a identificação dos elementos transversais e seus relacionamentos com os elementos-base. A aSideML tem um importante papel nesta etapa, pois seus diagramas geram artefatos capazes de comunicar de forma clara o papel dos aspectos, seus relacionamentos e características. As etapas de Teste e Implantação não serão abordadas neste trabalho.

5.1 Análise de Requisitos

A etapa da Análise de Requisitos envolve a identificação e modelagem do domínio da aplicação. É nesta etapa que são identificados os requisitos funcionais e não-funcionais. Esses requisitos são então mapeados e modelados como Casos de Uso. Por fim, são gerados os Modelos Conceituais de Negócio e identificados os casos de uso candidatos a serem implementados como aspectos.
Algumas atividades, como a identificação dos casos de uso, construção do diagrama de casos de uso e descrição dos casos de uso, acontecem de forma individual para os requisitos funcionais e não-funcionais do sistema. Posteriormente, alguns destes elementos são unidos em um diagrama de casos de uso. Uma visão parcial da união dos casos de uso funcionais e não-funcionais pode ser visto na figura 4. A figura apresenta três atores, a saber: Atendente, Gerente e Sistema de Faturamento. Esses atores possuem relacionamentos com casos de uso. Alguns casos de uso, por sua vez, possuem relacionamentos de extensão e inclusão com outros casos de uso. Existe ainda o caso de uso não-funcional “Registrar Operações”, que é identificado pelo estereótipo <<NFR>>, e possui um relacionamento de extensão com outros casos de uso.

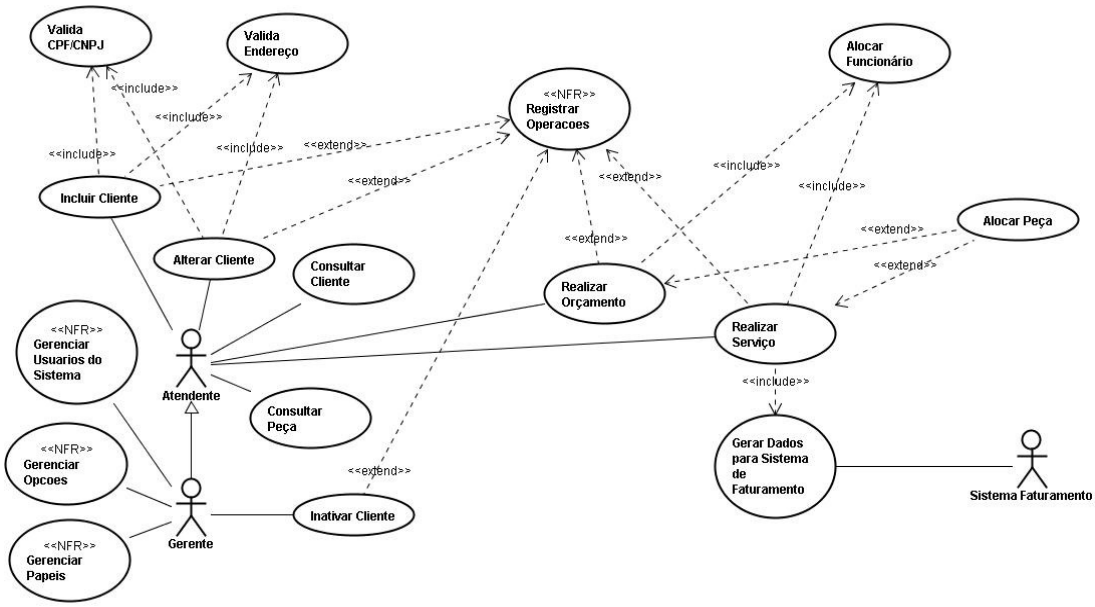


Figura 4: Visão parcial da união de casos de uso funcionais aos não-funcionais.

A última atividade desta etapa envolve a identificação dos primeiros candidatos a aspectos. Os casos de uso candidatos a serem implementados como aspectos são aqueles incluídos, estendidos ou restringidos, por dois ou mais casos de uso. Segundo [5], esse critério se justifica pois, pensando em

nível de implementação, esses casos de uso representam interesses espalhados no sistema. A tabela 2 apresenta os candidatos a aspectos identificados para o sistema em questão.

5.2 Especificação

A etapa de Especificação possui elevada importância no processo de desenvolvimento, visto que é nesta etapa que os componentes e aspectos são especificados para, em uma etapa posterior, serem providos [5].

Tabela 2: Candidatos a aspectos para o sistema para prestadores de serviço.

Caso de Uso	Tipo	Critério
Valida CPF/CNPJ	Funcional	Incluído por dois ou mais casos de uso
Valida Endereço	Funcional	Incluído por dois ou mais casos de uso
Alocar Peça	Funcional	Estende dois ou mais casos de uso
Alocar Funcionário	Funcional	Incluído por dois ou mais casos de uso
Autenticar Usuário	Não-Funcional	Estende dois ou mais casos de uso
Registrar Operações	Não-Funcional	Estende dois ou mais casos de uso
Controlar Acesso dos Usuários	Não-Funcional	Estende dois ou mais casos de uso
Persistir Dados	Não-Funcional	Estende dois ou mais casos de uso

Assim como as demais, esta etapa é compreendida por uma série de atividades. As duas primeiras atividades são relativas aos componentes base e suas interfaces. Essas atividades compreendem a identificação dos componentes e suas interfaces, bem como a interação entre esses componentes. Como resultado destas duas atividades, é produzido um diagrama com a arquitetura parcial dos componentes. A figura 5 apresenta a arquitetura parcial do sistema para prestadores de serviços. Esta figura apresenta apenas a arquitetura dos requisitos funcionais do sistema.

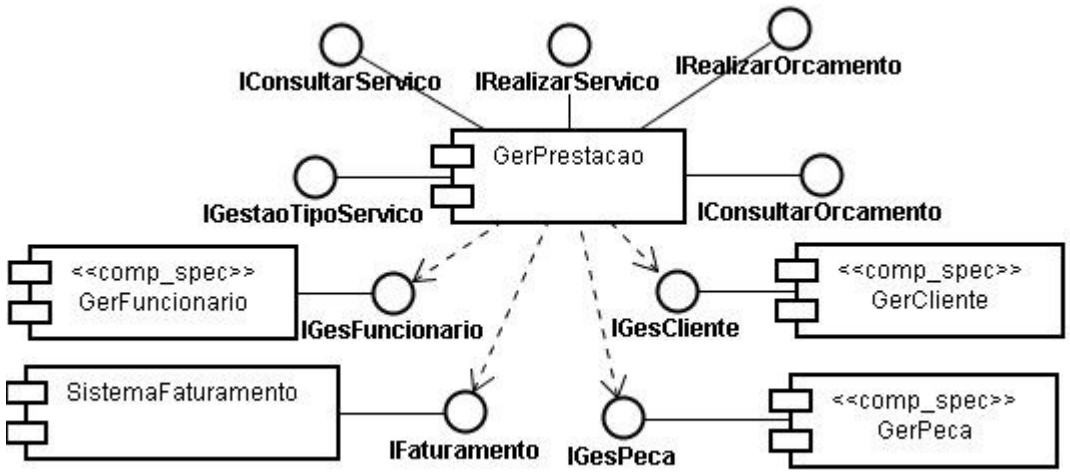


Figura 5: Arquitetura parcial dos componentes funcionais.

5.2.1 Aplicação da aSideML

Segundo o que foi descrito pelo método para DSBC/A, a próxima atividade é a identificação dos componentes transversais. Esta atividade é composta por algumas outras atividades, cujo objetivo é identificar os elementos que serão implementados como aspectos, elaborar suas interfaces e operações e, por fim, definir uma arquitetura para componentes base e transversais. A atividade

seguinte envolve a interação entre componentes-base e transversais. A linguagem aSideML foi aplicada na modelagem destas atividades, permitindo uma maior riqueza de detalhes na apresentação dos modelos.
Os critérios de Atomicidade, Disparo, Integridade, Variabilidade e Custo/Benefício são apresentados por [5] como critérios norteadores que, quando aplicados sobre os casos de uso candidatos a aspectos, auxiliam o analista e decidir pela implementação com aspectos. A tabela 3 apresenta os candidatos a aspectos, os critérios que cada um deles atende e a opção de implementação para cada um deles.
Para os casos de uso que não serão implementados como Aspectos, cabem ainda duas alternativas: ser implementado como uma operação interna do componente ou virar um novo componente.

Tabela 3: Candidatos a aspectos e a decisão de implementação para cada um.

Caso de Uso	Atomicidade	Disparo	Integridade	Variabilidade	Custo/Benefício	Implementação
Valida CPF/CNPJ	X	X	X		X	Aspecto
Valida Endereço	X	X	X		X	Aspecto
Alocar Peça	X	X				Outro
Alocar Funcionário	X	X				Outro
Autenticar Usuário	X	X	X		X	Aspecto
Registrar Operações	X	X	X		X	Aspecto
Controlar Acesso dos Usuários	X	X	X		X	Aspecto
Persistir Dados	X	X				Outro

Identificados os elementos que serão implementados como aspectos, utilizamos um diagrama de aspectos da linguagem aSideML para modelar os aspectos e suas interfaces. A figura 6 apresenta o diagrama de aspectos, na visão completa, para o aspecto "Registrar Operações". É importante observar ainda que na aSideML, um aspecto pode ser apresentado em sua visão completa ou condensada. Nesta figura, o aspecto GerRegistroOp possui uma interface chamada ITRegistrarOperacao. Esta interface transversal possui como característica uma operação de *crosscutting* registrarOperacaoExecutada(), que será utilizada como um refinamento a ser aplicado após a execução de uma determinada operação no componente base. O relacionamento de *crosscutting* de um aspecto com um componente base é apresentado no diagrama de classe estendido, que pode ser visto na figura 7.

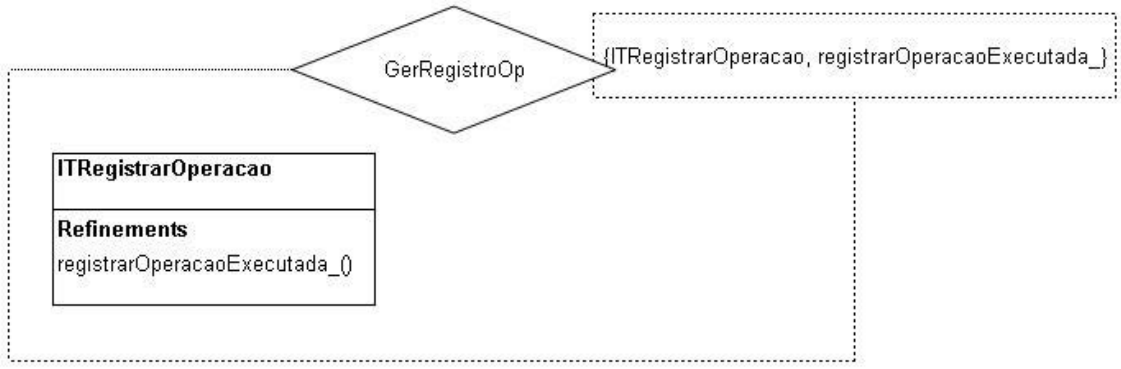


Figura 6: Diagrama de aspecto para o aspecto GerRegistroOp.

Para facilitar a visualização dos relacionamentos de *crosscutting*, [5] apresenta uma atividade que envolve a elaboração de novas interfaces para operações entrecortadas. Nesta atividade, é sugerido que todas as operações entrecortadas por um mesmo aspecto, sejam unidas em uma única interface, desta forma, é possível saber quais operações um aspecto entrecorta. Para exemplificar, vamos considerar as interfaces IRealizarOrcamento e IRealizarServico. A tabela 4 apresenta as operações destas interfaces e sinaliza aquelas que devem ser entrecortadas pelo aspecto GerRegistroOp.

Seguindo o que está descrito pelo método, as operações que são entrecortadas por GerRegistroOp deveriam ser unidas em uma única interface, como uma interface IEntradaDeDados, por exemplo.

Tabela 4: Análise para o agrupamento de operações entrecortadas – suprido pela aSideML.

Interface	Operações	Entrecorte por GerRegistroOP
IRealizarOrcamento	adicionarOrcamento	SIM
	consultarOrcamento	NÃO
IRealizarServico	adicionarServico	SIM
	consultarServico	NÃO
	alterarServico	SIM
	inserirPecas	SIM

O diagrama de classe estendido da aSideML pode ser utilizado para modelar os relacionamentos de *crosscutting*, tornando esta atividade desnecessária. Esse diagrama apresenta o relacionamento de *crosscutting* em nível estático entre um aspecto e um elemento base. O diagrama apresenta notação específica para cada relacionamento, tornando possível a identificação de qual operação será entrecortada por cada operação no aspecto. A figura 7 apresenta o diagrama de classe estendido para a interface IRealizarOrcamento. O relacionamento de *crosscutting* é identificado pelo estereótipo <<crosscutt>>, seguido pela operação transversal do aspecto e a operação que será entrecortada no elemento base. Nessa representação, o aspecto é apresentado em sua visão condensada.

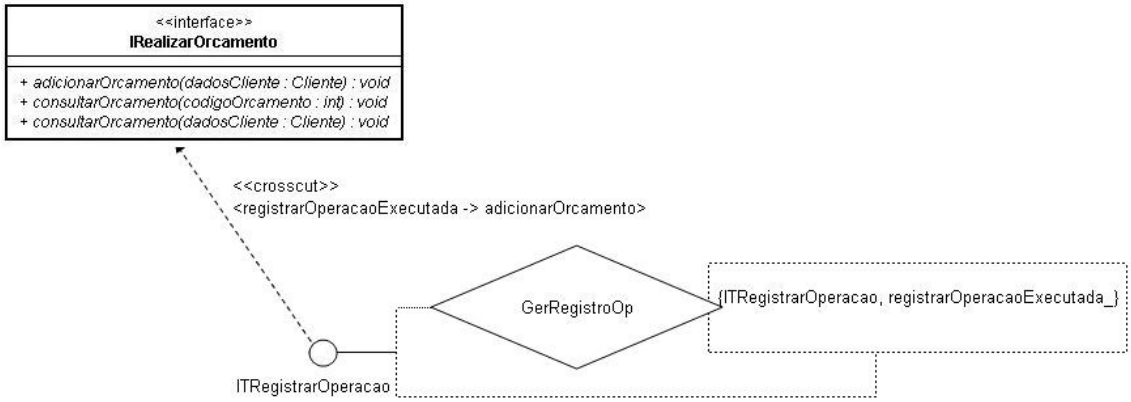


Figura 7: Relacionamento de *crosscutting* entre aspecto e interface.

Alguns diagramas da aSideML oferecem diferentes perspectivas. A figura 8 apresenta uma visão com a perspectiva centrada em ponto de combinação. Nessa perspectiva é possível observar um conjunto de elementos base e quais das suas operações são entrecortadas por determinados aspectos. Uma das últimas atividades é a interação entre componentes base e transversais. Essa interação foi modelada através dos diagramas comportamentais da aSideML

5.3 Provisionamento e Montagem

Nas etapas anteriores, foram criadas as estruturas necessárias para a geração do sistema. Nesta etapa, os componentes que satisfazem as condições especificadas são providenciados e unidos, formando uma aplicação completa. No desenvolvimento baseado em componentes, existem três alternativas de provisionamento. A primeira consiste no reuso de componentes utilizados em outras aplicações, a segunda alternativa é aquisição de componentes de terceiros e, por fim, a implementação de novos componentes.

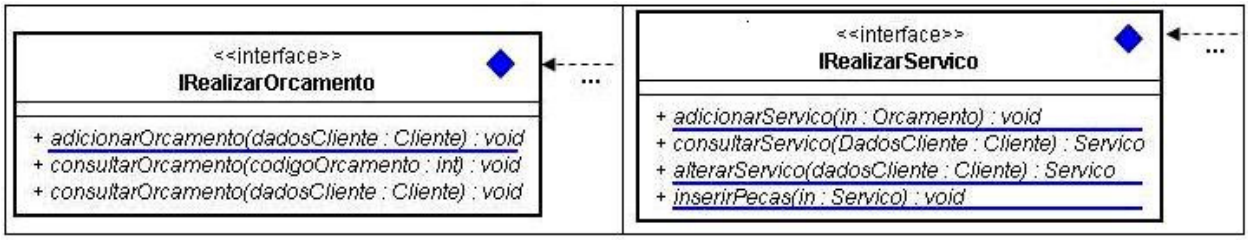


Figura 8: Visualização de *crosscutting* – perspectiva centrada em ponto de combinação.

Para este trabalho, alguns componentes foram adquiridos através de repositórios de componentes na Web e, para aqueles não disponíveis, houve a implementação. Já para os aspectos, todos foram implementados utilizando a linguagem AspectJ.

Para permitir que os aspectos implementados para esta aplicação pudessem ser reutilizados, estes foram implementados de forma genérica. Isso significa que suas operações e interfaces são definidas, porém, os pontos de entrecorte serão definidos apenas no momento da montagem dos componentes e aspectos, através do uso de conectores [5]. A figura 9 apresenta dois quadros. No lado esquerdo, é possível observar uma transcrição genérica para a implementação do aspecto GerRegistroOp. Observe que o aspecto é declarado como abstrato e suas operações internas são definidas. Vale ressaltar que, como a linguagem AspectJ não implementa interfaces, o conjunto de pontos de junção recebe o nome da interface.

Na etapa de Montagem, os componentes devem ser unidos, e a interface com o usuário, implementada. Neste momento, os conectores que unirão os aspectos aos seus pontos de entrecorte, também devem ser desenvolvidos. Na figura 9, o quadro à direita apresenta um conector para o aspecto GerRegistroOp. Este conector é uma extensão do aspecto GerRegistroOp, declarado anteriormente como abstrato. Neste conector, o ponto de entrecorte, antes declarado apenas genericamente, é sobrecarregado por um ponto específico.

```
abstract aspect GerRegistroOp {
        public abstract pointcut ITRegistrarOperacao();
        public registrarOperacaoExecutada () {
                ...
        }
        after(): ITRegistrarOperacao() {
                registarOperacaoExecutada();
                ...
        }
}
```

```
public aspect conectorParaGerRegistroOp
        extends GerRegistroOp {
        public pointcut ITRegistrarOperacao():
                call (* *.adicionarOrcamento (..));
}
```

Figura 9: Implementação de aspecto genérico e conector.

6 CONCLUSÕES

O método para DSBC/A fornece critérios úteis para a identificação e implementação de aspectos durante o processo de desenvolvimento de *software*. As atividades contidas no método fornecem uma forte documentação para os componentes e aspectos, possibilitando o reuso e facilitando a manutenção. A combinação da aSideML para modelagem da especificação de aspectos, suas características e relacionamentos com componentes base, gera modelos de maior entendimento, além de evitar algumas atividades adicionais, propostas anteriormente pelo método. Os modelos gerados pela aSideML explicitam de forma simples e completa as características da orientação a aspectos. A combinação destas técnicas tornou possível o desenvolvimento de uma aplicação

completa baseada em componentes e aspectos, onde ambos são tratados como cidadãos de primeira classe, podem ser reutilizados e facilmente atualizados.

REFERÊNCIAS

[1] Bezerra E. Princípios de Análise e Projeto de Sistemas com UML. Elsevier, Rio de Janeiro, 2002.

[2] Booch G. UML: Guia do usuário. Campus, Rio de Janeiro, 2000.

[3] Chavez C. V. F. G. "Um Enfoque Baseado em Modelos para o Design Orientado a Aspectos," tese de doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2004.

[4] Cheesman, J. e Daniels J. UML Components: A simple process for specifying component-based software. Addison-Wesley, 2000.

[5] Eler M. M. "Um método para o desenvolvimento de software baseado em componentes e aspectos," dissertação de mestrado, Universidade de São Paulo, São Carlos, 2006.

[6] Garcia A. et al. *Relatório do Primeiro Workshop Brasileiro de Desenvolvimento de Software Orientado a Aspectos*, tech. report, 18º. Simpósio Brasileiro de Engenharia de Software, 2004.

[7] Garcia A. e Chavez C. V. F. G. "Desenvolvimento de Software Orientado a Aspectos", *19º. Simpósio Brasileiro de Banco de Dados e 18º. Simpósio Brasileiro de Engenharia de Software*, Brasília, 2004.

[8] Gimenes I. e Huzita E. H. M. Desenvolvimento Baseado em Componentes: Conceitos e Técnicas. Ciência Moderna, Rio de Janeiro, 2005.

[9] Junior V. G. S. e Winck D. V. e Machado C. A. P. "Programação Orientada a Aspectos Abordando Java e AspectJ", *União de Tecnologia e Escolas de Santa Catarina*, Joinville, [2003?].

[10] Laddad R. AspectJ in Action – Practical Aspect-Oriented Programming. Manning Publications Co., Connecticut, 2003.

[11] Silva G. C. et al. "Uma Introdução a Programação Orientada a Aspectos", CD-ROM, *VIII Semana de Informática e V Mostra de Iniciação Científica do Curso de Sistemas de Informação*, Paranavaí, 2006.

[12] Sommerville I. Engenharia de Software. Addison-Wesley, São Paulo, 2003.

Crítica a los cursores de SQL y propuesta de su eliminación

Ing. Hugo Ryckeboer, Ing. Osvaldo Spositto, Ing. Alfonso Palomares
Departamento de Ingeniería e Investigaciones Tecnológicas,
Universidad Nacional de La Matanza,
Florencio Varela 1903
B1754JEC San Justo, Argentina
cursores@unlam.edu.ar

Abstract

Using an exhaustive analysis of possible program structures we show that SQL cursors are not necessary to compute tables. We highlight the incompatibility between cursors and the descriptive and functional syntax style. The use of cursors also destroys the potential parallelism in execution. We show how arbitrary restrictions in the structure of the SQL language bound the languages expressive power.

Key words: data bases, cursors, programming theory, ANSI-sql,

Resumen

A través de un análisis exhaustivo de estructuras de programas se muestra que los cursores de SQL son innecesarios para elaborar tablas. Se resalta la incompatibilidad entre los cursores y el estilo descriptivo y funcional de la sintaxis y que su uso destruye el potencial paralelismo en la ejecución. Se señalan restricciones arbitrarias a la generalidad que limitan el poder expresivo del lenguaje SQL.

Palabras claves: bases de datos, cursores, teoría de programación, ANSI-sql,

1. INTRODUCCIÓN

El lenguaje SQL no sólo administra y recupera datos almacenados sino que a través de cierta capacidad de cálculo y de especificación de procesos ofrece soluciones casi integrales, eliminando en muchos casos el uso de lenguajes de programación tradicionales. El hecho de especificar lo que se pretende y no el modo detallado de obtenerlo implica un salto conceptual y permite calificarlo con toda propiedad como uno de los así llamados lenguajes de cuarta generación.

Por otra parte el motor de ejecución puede introducir un alto grado de paralelismo en la operación, característica que las construcciones de los lenguajes de programación habituales no ofrecen o las que hay no son sencillas de usar. También a través de las subconsultas se incorpora a su concepción descriptiva un estilo funcional, que expresa lo que desea sin efectuar asignaciones intermedias.

Es difícil decir que es SQL en este momento, tomamos como referencia ese común denominador de todas las implantaciones, inspirado en el SQL92. Ese SQL no ofrece una solución total por insuficiencia en los recursos que provee. Por otra parte, ha incorporado un recurso, los cursores, que no armoniza con su paradigma fuertemente inclinado al paralelismo en la ejecución.

Muchos tutoriales [1,4] y suscriptores a listas dedicadas al SQL, desaconsejan el uso de cursores, sin aportar un fundamento.

En este trabajo pretendemos fundamentarlo, mostrar cómo evitarlos sistemáticamente, delimitando los pocos casos en los cuales su uso es actualmente inevitable y señalar las pocas construcciones que debieran agregarse al SQL para hacer de él un lenguaje completo, en el sentido de que sea capaz de expresar lo mismo que los lenguajes imperativos habituales. Si esto se hiciera se aumentaría simultáneamente la capacidad expresiva del lenguaje en su modalidad descriptiva y la eficiencia en la paralelización de las operaciones.

Finalmente señalaremos las líneas nuevas de investigación que hemos detectado como fruto de este análisis.

2. PROGRAMAS QUE OPERAN SOBRE TABLAS.

El paradigma de SQL está orientado a una visión relacional de los datos y de los resultados. Toda operación describe la elaboración de una tabla a partir de las tablas preexistentes. Como caso extremo se elabora una sola tupla y eventualmente a ésta conteniendo un solo campo.

En esa visión relacional se disponía de las operaciones básicas y algunas derivadas del álgebra relacional: selección, proyección y ensamble. Rápidamente se incorporó una capacidad de cálculo, expresiones entre componentes de columnas y operaciones sobre los elementos de una columna, con lo cual muchas necesidades podían ser resueltas íntegramente dentro del lenguaje SQL. Mucho se tardó en incorporar funciones definidas por el usuario, difícil de entender porqué, puesto que compilar/interpretar un trozo de programa no es mucho más complejo que hacerlo con una expresión arbitraria.

En este contexto, los cursores ofrecen al programador la posibilidad de recibir una tupla por vez y someterla a las operaciones que considera conveniente. Si esto ocurre dentro del contexto de un lenguaje de programación incorpora toda la capacidad expresiva de este último.

Consideramos que la serialización de las operaciones que introduce el cursor es perjudicial y nos proponemos poner en evidencia que existe un camino alternativo mucho más conforme al espíritu del SQL, conservando la predisposición para un elevado paralelismo y el estilo funcional propio del lenguaje.

Como método para demostrar la no necesidad de los cursores, hemos partido de una taxonomía de las posibles acciones que se pueden realizar sobre una tabla. Las hemos caracterizado al modo del paradigma imperativo. Su modo de operar sobre tablas serían las iteraciones. De ser traducidos literalmente hubieran introducido cursores. Luego ponemos en evidencia que la capacidad expresiva del SQL no los necesita. Después estudiamos las combinaciones de las acciones elementales y su traducción al SQL. Suponemos que el programador puede describir las tablas sobre las cuales opera recurriendo a las construcciones SQL que corresponden al álgebra relacional. A partir de esa tablas mostraremos cómo programaría un programador con mentalidad secuencial propia del paradigma imperativo. Mostraremos por un análisis exhaustivo de casos como es factible realizar siempre la conversión siempre y cuando el SQL reciba un enriquecimiento que oportunamente destacaremos. Versiones posteriores al SQL92 han recogido algunas de ellas y fabricantes de motores han tomado antes la iniciativa en ese sentido.

Al convertir programas iterativos a SQL no incluiremos consideraciones de eficiencia por cuanto consideramos que éstas se pueden delegar en los algoritmos de optimización de los motores de bases de datos. También en esta versión nos limitamos a servicios de consulta que si bien fabrican tablas finales o intermedias no modifican las originales. Las actualizaciones se pueden conceptualizar como construcción de tablas nuevas que modifican o sustituyen a las recibidas.

Programas que producen resultados visibles, en pantalla o impresos se pueden descomponer en la preparación del material en forma de una tabla seguido de un proceso que la exterioriza.

También se puede entender que se fabriquen tablas intermedias, por ejemplo una selección compleja puede elaborar primero las claves de los elementos de interés operando sobre los atributos que influyen en la selección y luego seleccionar a estos con todos sus atributos.

Es por ello que supondremos para asemejarnos a una consulta SQL que se parte de una tabla existente (tal vez una vista) y se culmina con otra, la cual, es utilizada en otra etapa o una vez maquillada, se exhibe. En una primera visión podríamos decir que disponemos de variables que almacenan tablas y cada proceso asigna una nueva tabla y finalmente mostraremos que esas tablas

intermedias no deben ser citadas explícitamente ya que SQL las puede incorporar a modo de subconsulta. Nuestro análisis comienza con una taxonomía de los procesos sobre tablas.

3. TAXONOMÍA DE LOS PROCESOS

El hecho de ser las tablas conjuntos restringe enormemente las operaciones que se puede hacer con uno de sus elementos. Toda acción que privilegia a algunos elementos respecto de otros rompe la homogeneidad intrínseca del conjunto y equivale a particionar ese conjunto, o sea esa tabla, en partes, de modo tal que en cada parte se respete la homogeneidad de los elementos.

Bajo esta óptica resaltan dos grandes categorías de proceso: Por un lado aquellos que calculan a partir de los atributos de un elemento atributos nuevos. Esta misma acción se puede repetir en cada elemento y por lo tanto preservan la homogeneidad en el procesamiento. Estos procesos los denominaremos *horizontales*.

Luego están los que elaboran un resultado representativo de todos sin interesar el orden en que se los suministre, o sea, que evalúan una función simétrica de los datos. Sin embargo merecen ser subdivididos por cuanto existen funciones que son generalización a conjuntos de operaciones binarias, el ejemplo más sencillo es la sumatoria. Estos cálculos hacen intervenir una sola vez a cada dato en cualquier orden y pueden computarse a partir de resultados parciales sobre subconjuntos del conjunto base,. A estas clase de procesos los denominaremos *verticales*, a los restantes, o sea , a los que implican múltiple operación sobre al menos una parte de los datos, *algorítmicos*. Estos últimos tienen apariencia de dar un trato desigual a los elementos. Esta desigualdad es sólo aparente. Con razonamientos matemáticos se puede demostrar que el resultado no depende de las decisiones arbitrarias por las cuales le dieron a ciertos elementos un mayor protagonismo que a otros. Si los protagonistas hubieran sido otros, el resultado final no hubiera cambiado.

3.1 Procesos horizontales

Teniendo a la vista un elemento (una tupla) puede decidir extenderla con campos adicionales. Estos cálculos hacen intervenir a los campos ya conocidos. También incluiremos aquí consultas con unicidad a otras tablas. En su forma más sencilla estos cálculos se expresan como expresiones arbitrarias con operadores propios del tipo de datos que intervienen en la misma. A nivel de SQL92 se dispone de alternativas con el CASE. No hay constructor de iteraciones, ni forma de tener una biblioteca de cálculos frecuentes. Hay empresas que proveen motores de bases que ya subsanaron esta limitación.

Dado el carácter de conjunto que tiene una tabla, no hay forma de distinguir entre sus elementos, por lo cual, lo que hace con uno de ellos lo debe hacer con todos por igual, esta afirmación no queda invalidada por el hecho de que el proceso repetido sobre todos los elementos contuviera alguna bifurcación. Por otra parte el cálculo hecho para una tupla no interfiere con el cálculo hecho para otra. Aquí se destaca el elevado paralelismo que puede haber en los procesos horizontales.

3.2 Procesos verticales

En estos utiliza los elementos para calcular un valor representativo de todo el conjunto. En cálculos sencillos esto se consigue inicializando previamente una variable la cual es actualizada con cada tupla procesada. Así por ejemplo la sumatoria de los valores de un campo se construye por sucesivas sumas en las cuales es argumento y destino la variable que contendrá al final la sumatoria buscada.

Este tipo de cálculo se presta para su distribución, así una sumatoria se puede elaborar a partir de dos sumatorias parciales. Esta característica no es casual, al tratarse de un conjunto, sólo tienen una clara semántica las operaciones simétricas respecto de sus operandos. Esa simetría surge de la conjunción de las propiedades asociativas y conmutativas, característica que tienen las funciones provistas: MAX, AVG… En su versión más sencilla, tal como está implantado en SQL92 interviene un solo atributo, pero no se ve motivo para imponer esta limitación. En operaciones más complejas

el resultado podría ser una tupla y no un valor aislado como sería un cálculo de ejes de inercia de un cuerpo descrito mediante un conjunto de masas posicionadas.

3.3 Procesos algorítmicos

Almacenar la totalidad de los valores de ciertos campos con un objetivo similar al anterior pero delegado en un algoritmo de procesamiento no lineal o al menos no secuencial de los valores. Dado la indiferencia del algoritmo frente al orden de los elementos en su primer ciclo recorre los elementos secuencialmente. Luego algunos valores son consultados reiteradamente. El mayor protagonismo que dan a algunos elementos es sólo aparente. Se puede demostrar que el resultado final no depende de cual haya sido el protagonista. Cuando se trata de algoritmos de orden los razonamientos usan preponderantemente la propiedad transitiva de las relaciones de orden. Un ejemplo más algebraico es la inversión de una matriz donde en cada paso se elige un elemento como "pivote".

Son ejemplos:

- El cálculo de la mediana de una columna que entrega uno o dos valores.
- El cálculo de una cápsula convexa procesa dos o más columnas. Devuelve una cantidad variable de los elementos recibidos, desde unos pocos hasta la totalidad.
- La inversión de una matriz entrega una tabla de igual orden y cardinalidad pero con un campo numérico cambiado.

4. ANÁLISIS DE LOS PROGRAMAS QUE OPERAN SOBRE TABLAS

En los próximos párrafos consideraremos con un alto grado de abstracción los procesos enumerados y sus combinaciones. Los detalles del cálculo que se estuviera realizando no son de interés en esta clasificación. Hablaremos de funciones que de tuplas calculan tuplas. Es usual que tales funciones se puedan describir componente a componente mediante funciones de tuplas en campos elementales. El querer conservar algún campo de la tupla recibido se reduce a agregar una componente más en la tupla resultado que se construya por copia de un campo de la tupla original. Usaremos letras minúsculas empezando con la "`f`" para referirnos a tales funciones.

Por ejemplo si escribimos:

```
SELECT   x as a, x * y + z as b
```

Estamos construyendo una tupla de dos componentes, `a` y `b`, a partir de una de tres componentes, `x`, `y` y `z`. Podríamos escribir `a, b ← f( x, y, z )` o, si no queremos entrar en tanto detalle, `s ← f( t )` siendo `s` y `t` nombres genéricos de tuplas.

Se puede observar en el ejemplo que la componente `a` es una trascripción de `x`

El SQL92 impone una restricción a las funciones que se pueden calcular de este modo, permitiendo utilizar en su elaboración solamente ciertas funciones y operaciones predefinidas y un mecanismo de selección de alternativas.

Los proveedores de motores ya descubrieron que esa limitación es arbitraria y perjudicial para la expresividad del lenguaje y permiten escribir funciones utilizando también ciclos.

Aún así, sigue habiendo una limitación sintáctica, por cuanto las funciones provistas no pueden aportar varios campos de una vez. Un ejemplo práctico de tal necesidad sería una función que cambie el sistema de coordenadas en un problema geométrico.

En lo que sigue usaremos las siguientes convenciones:

$\|$ es el operador de concatenación de tuplas

$\bowtie$ es el operador de ensamble, llamado también de junta, ("`JOIN`")

Suponemos que las operaciones del álgebra relacional: selección, proyección, producto cartesiano, ensambles,… son operaciones disponibles en el motor de base de datos. Para no introducir más símbolos consideraremos el producto cartesiano como caso extremo del ensamble.

Con la idea amplia de función explicitado más arriba, la proyección se reduce a una función que consista en escribir sólo los nombres de los atributos que sobreviven.

La selección la consideramos una operación fundamental para evitar el envío de volúmenes demasiado grandes hacia el programa. Bien utilizada evitará el uso de selecciones dentro de la aplicación. Para que estas selecciones sean efectivas y evitar pasos intermedios es conveniente tener la misma capacidad de usar funciones arbitrarias en el `WHERE` como en el `SELECT`. Ambos remiten en su sintaxis a `<value expression>`, asegurando por lo tanto la igual capacidad expresiva.

El ensamble, simbolizado con $\bowtie$ es una operación si bien derivada fundamental en cuanto al poder expresivo, la cual por otra parte no hace más que revertir las proyecciones hechas a partir de una relación universal. El ensamble aumenta el tráfico de datos entre servidor SQL y cliente. Es de notar que un programador de tuplas en un lenguaje imperativo evita transferencias repetidas de partes comunes de las tuplas efectuando el ensamble en la memoria principal ocupada por el programa. Esto lo señalaremos nuevamente más adelante.

El lenguaje SQL provee operaciones de conjuntos. Las señalaremos genéricamente con mayúsculas `F`; `G`,… Aquí también señalamos una pobreza del lenguaje SQL. El usuario debiera ser capaz de definir nuevas funciones de conjunto e inclusive permitir algunas de resultado no atómico o que operen sobre varias columnas al mismo tiempo (por ejemplo cálculo de regresiones)

Nos imaginamos una sesión de SQL como una sucesión de acciones que de tablas engendran una nueva tabla. Sus variables, para almacenar resultados intermedios son del tipo tabla. No nos interesa si dos tablas tienen parte de su cálculo en común, lo consideramos problema del optimizador del motor.

En los esquemas siguientes ilustramos el modo de operar de un programador en un lenguaje imperativo que quisiera construir una tabla a partir de otra. Introducimos un iterador no común en los lenguajes de programación: `foreach.` Con el operador `foreach` queremos resaltar la indiferencia frente al orden de evaluación. Justamente esto es lo que provee un proceso con cursores, acceso a un elemento por vez en un orden arbitrario. Que la tabla o vista que alimenta al cursor entregara los elementos en un orden particular no tiene ninguna importancia para quien quiera construir otra tabla, solamente serviría para una exhibición más inteligible de la misma.

Comenzamos el análisis con los procesos elementales para luego estudiar combinaciones de los mismos. Si llamamos **D** a la tabla dato y **R** a la tabla resultado, tendríamos las siguientes estructuras para los distintos procesos

4.1 Procesos elementales

Usamos la taxonomía del punto3 y damos sus equivalentes en SQL sin utilizar cursores.

4.1.1 Proceso de tipo horizontal

```
R ← ∅
foreach t of D do
    s ← f( t )
    agregar s a R
```

```
SELECT  f( t )
FROM  D
```

Esto admite una traducción directa a SQL si no tuviera limitación en las funciones.:

La `f( t )` se traduce tal como indicamos más arriba. Si `R` no es la pantalla falta una cláusula `INTO R` .Recuérdese que dijimos que `D` puede ser fruto de una expresión del álgebra relacional y por lo tanto constar de la cita de varias tablas e incluir cláusulas `WHERE` y `JOIN` o inclusive ser una subconsulta.

4.1.2 Proceso de tipo vertical

```
Inicializar r
foreach t of D do
    r ← f( t, r )
```

El resultado R se reduce aquí a una única tupla r.

Esto admite en muchos casos una traducción directa a SQL:

```
SELECT  F( t )
FROM  D
```

Aquí tenemos la principal limitación del SQL, el repertorio de las F está delimitado en la definición de lenguaje y no hay un mecanismo de definición de nuevas instancias. Esta pobreza se ha paliado agregando en algunas implantaciones nuevas funciones predefinidas, pero éste no es el camino correcto, debiera haber un mecanismo de extensión. Este es el principal defecto en el diseño del lenguaje. Es notable como en las versiones más modernas, incluyendo la versión de trabajo del SQL-2003 [3], se persiste en el mismo defecto, habiendo puesto como remedio un surtido más grande de funciones predefinidas.

La pobreza de funciones de conjuntos es la principal causa del uso actual de cursores.

Después de este diagnóstico, y para no reiterarlo, continuaremos el análisis suponiendo que tenemos un mecanismo para definir nuevas funciones y que esta limitación está superada.

4.1.3 Proceso de tipo algorítmico

```
n ← 0 ;
foreach t of D do
    n ← n + 1
    v[n] ← t
R ← Algoritmo( v, n )
```

SQL no ofrece una solución genérica. Tampoco los cursores. Hay versiones de SQL que permiten depositar el contenido íntegro de una tabla en una variable de tipo tabla. Después se debiera llamar a una función o procedimiento ajeno al SQL para que elabore un algoritmo.

El primer inconveniente de este modo de operar es la rotura de la estructura funcional.

Los algoritmos que apartamos en esta tercera variante consultan el vector v en una forma no secuencial. El cálculo de mediana es un ejemplo, entrega una o dos tuplas de un sólo campo sobre un R que consta de un sólo campo.

Aunque hay métodos lineales, éstos tienen coeficiente mayor que 1, lo que significa que algunos elementos son consultados varias veces como consecuencia de la evaluación parcial ya hecho además de ser desplazados. El direccionamiento directo es fundamental para lograrlo. Estos problemas no se encaran con cursores sino tal como muestra el esquema con ayuda de un vuelco en memoria de la totalidad de la información disponible. El depósito masivo de toda una tabla es mucho más eficiente que llamar sucesivamente a la función FETCH NEXT del cursor con el único propósito de depositarla en un arreglo. Ya se hizo notar que estos algoritmos en su primera fase hacen un barrido secuencial de los datos y recién en sus posteriores accesos actúan selectivamente.

La solución ideal sería que SQL permitiera incorporar funciones que de tablas den tablas. No queda tan clara la utilidad cuando hay un único cálculo algorítmico, pero cuando este sea repetitivo, su incorporación al lenguaje permitiría distribuir las diferentes ejecuciones. Fuera del cálculo de mediana no es frecuente este tipo de situaciones entre los usuarios del SQL.

4.2. Parametrización de Procesos

Es factible que estos procesos dependan de un parámetro. Esto se aplica particularmente cuando un proceso está anidado dentro de otro. El contenedor puede, al recorrer su propia tabla, enviar un parámetro al contenido

En los procesos horizontales el parámetro puede intervenir en uno o ambos modos siguientes:

(a) La especificación de D (b) El modo de efectuar el cómputo de s.

El parámetro estará simbolizado con p.

En el primer uso, el parámetro especifica el conjunto particular sobre el cual se quiere operar. Interpretado el parámetro cómo especificador de una tabla,

4.2.1 Proceso horizontal parametrizado

```
R ← ∅
foreach t of D( p )
do
  s ← f( t )
  grabar s en R
```

```
R ← ∅
foreach t of D do
  s ← f( t, p )
  grabar s en R
```

También se puede dar combinado.

4.2.2 Proceso vertical parametrizado

En los procesos de tipo vertical el parámetro puede intervenir solamente en la especificación de D

```
Inicializar r
foreach t of D( p ) do
  r ← f( t, r )
```

Cuesta imaginar que intervenga en el cómputo. Por ejemplo ¿Cómo sería una sumatoria parametrizada (y no en la cantidad de sumandos)? De todos modos no molestaría que lo hiciera.

4.2.3 Proceso algorítmico parametrizado

Aquí podríamos imaginar un proceso como el cálculo de percentiles que admite el valor de éste como parámetro. Pero un algoritmo que los calculara simultáneamente sería siempre más eficiente que reiterados cálculos de valores únicos.

La especificación del conjunto de datos sobre el cual correr el algoritmo es otro uso del parámetro.

```
n ← 0 ;
foreach t of D( p ) do
    n ← n + 1
    v[n] ← t
R ← Algoritmo( v, n, p )
```

4.3 Comentario sobre la trama de los procesos

Es de notar que hablando con esta generalidad en estos procesos elementales no hay instrucciones de selección, esto se puede razonar en detalle, por ejemplo para un proceso de tipo horizontal.

```
foreach t of D do
    s ← f( t )
    grabar s en R
```

Toda selección que afecta al cómputo de s es parte integrante de lo que genéricamente hemos llamado "f". Esto a lo más indica que debe haber un lenguaje algorítmico para describir estos cómputos. Una selección que influya en que s exista o en la cantidad de campos hace carente de sentido o imposible la existencia de un resultado para agregar a R.

Una selección que haga condicional la grabación hace también condicional el cálculo de s pues no tiene sentido calcular algo que no se piensa almacenar.

Finalmente si ambas líneas son condicionales resulta que la tabla D de partida tuvo que ser mejor descripta con una cláusula WHERE mejor especificada para que no aparezcan en ella tuplas no deseadas. El SQL es ortogonal en su <search condition>, de modo tal que lo que se puede especificar en una bifurcación también se puede especificar en un WHERE.

Sólo al componer procesos aparecen las acciones condicionadas. La selección aparece cuando el resultado de una consulta influye en que se realice o no la segunda consulta.

4.4 Anidamiento de procesos en ciclos

Un primer modo de anidamiento es tener un ciclo `foreach` dentro de otro. Si los valores del ciclo exterior no parametrizan a la tabla del ciclo interior se trata de un producto cartesiano de tablas, caso extremo del ensamble. Si por el contrario, si lo parametrizan se trata de un ensamble. El orden de los dos constructores de ciclos se podría invertir, adecuando la descripción de tabla parametrizada sin cambiar el resultado. Si se lo hace con un "`JOIN … ON …`" la condición expresada no cambia.

```
foreach t1 of D1 do
    s1 ← f1( t1 )
    foreach t2 of D2(s1) do
```

equivale pues a

```
foreach t of D1 ⋈ D2 do
```

Al poder parametrizar los procesos elementales arriba detallados resulta factible encerrar a cualquiera de ellos en un ciclo que provea sucesivos valores de parámetro para los mismos.

Esta y otras igualdades expresivas presuponen que toda función de la cual disponga el sistema se pueda usar tanto en la claúsula `SELECT` como en la cláusula `WHERE`.

4.4.1 Ciclos sobre procesos horizontales

Un ciclo que abarca un proceso horizontal produciría en cada iteración una tabla. Esto es incompatible con el planteo original de que el resultado sea una única tabla. El razonamiento de homogeneidad, en este caso aplicado al parámetro, deja pocas opciones: se debe reunir tablas en cantidad variable en una sola y que ello no dependa del orden de evaluación.

Operaciones asociativas y conmutativas entre conjuntos son la unión, intersección y diferencia simétrica. La última es de dudosa semántica aplicada a tuplas.

```
foreach t1 of D1 do
    R ← ∅
    foreach t2 of D2(t1) do
        s ← f1( t1 , t2 )
        agregar s‖t1‖t2  a R
    Aqui debo indicar que hago con cada
R
```

Si se trata de la unión basta con adelantar la sentencia `R ← ∅` un lugar, y la unión en el sentido que le da SQL es automática. Pero una vez sacada esta instrucción tenemos dos `foreach` anidados que según lo visto en 4.4 se confunden en 1. Si se trata de la intersección se está efectuando una división y esta es expresable en el álgebra relacional. Lo más general sería conservar los valores junto con los valores de los parámetros que le dieron lugar, comportamiento similar al de la unión.

4.4.2 Ciclos sobre procesos verticales

El planteo de partida es similar

```
foreach t1 of D1 do
    Inicializo r
    foreach t2 of D2(s1) do
        r ← f1( r, t1 , t2 )
    Aqui debo indicar que hago con cada
r
```

Valen las conclusiones anteriores si `r` se considera una relación de una sola tupla. El hecho de ser un escalar abre la posibilidad de efectuar con todos ellos un proceso vertical, caso que estudiaremos más adelante. Con la primera alternativa caemos en la construcción que en SQL usaría la cláusula `GROUP BY:`

```
R ← ∅
foreach t1 of D1 do
    Inicializo r
    foreach t2 of D2(t1) do
        r ← f1( r, t1 , t2 )
    guardar t1‖r en R
```

se traduce como

```
SELECT t1, F1( t1, t2 )
FROM  D1 JOIN D2 ON … t1 …
GROUP BY t1
```

4.4.2 Ciclos sobre procesos algorítmicos

Siguiendo los planteos anteriores:

```
foreach t1 of D1 do
    n ← 0 ;
    foreach t2 of D2( t1 ) do
        n ← n + 1
        v[n] ← t2
    R ← Algoritmo( v, n, p )
    Aqui debo indicar que hago con cada
R
```

Tiene una problemática similar a la de los procesos horizontales, se dispone de múltiples tablas y es necesario fijar un modo de reunirlos en una única tabla. Lo que si desaparece es la idea de reunir los dos ciclos, puesto que el ciclo interior no sería adyacente sintácticamente al exterior sino quedaría separado por la llamada a función que computa al algoritmo.

```
SELECT
FROM D1 JOIN Algoritmo( SELECT D2.t2 FROM D2 WHERE   D1.t1   )
```

En cambio con un proceso horizontal hubiéramos tenido:

```
SELECT
FROM  D1  JOIN  ( SELECT D2.t2  FROM D2 WHERE    D1.t1   )
```

Está última puede reescribirse sin el uso del segundo SELECT cómo ya hemos indicado.

Con cualquiera de estos procesos si en su realización dependieran de un parámetro, y los cómputos se ejecutan para varios valores del mismo, se crearía una tabla con un campo adicional dedicado al parámetro, el cual integrará la clave.

Actualmente los programadores depositan las tablas a ser procesadas algorítmicamente. La regularidad de las representaciones en memoria principal hace que esta área se pueda considerar un vector y a eso hemos aludido en nuestros esquemas. No molestaría que fueran listas u otro tipo de estructura.

El hecho de necesitar un almacenamiento para empalmar con un algoritmo escrito en un lenguaje procedural destruye el carácter funcional de las consultas. Lo que se necesita son funciones que de tablas den tablas.

4.5 Anidamiento de procesos en procesos

Los procesos que hemos considerado se esquematizan con un ciclo, a diferencia del caso anterior estos ciclos ya tienen un objetivo y lo que debemos destacar como aporte novedoso es de qué manera puede contribuir el ciclo interior al exterior.

4.5.1 Proceso horizontal dentro de otro horizontal

En un esquema que opera registro a registro corresponde que el ciclo interior colabore con el exterior en cada registro que elabora. El único modo es entregarlo tal como está para que sea incorporado sin más en el ciclo exterior o enriquecido por éste con elementos que el ciclo interior no elabora.

Arrancando desde una óptica distinta llegamos exactamente a lo que se hacía en el punto 4.4.1.

4.5.2 Proceso vertical dentro de otro horizontal

Aquí corresponde hacer una distinción: este proceso vertical se realiza antes o después del proceso horizontal. Expresamos esquemáticamente las dos posibilidades:

```
R ← ∅
foreach t1 of D1 do
    s1 ← f1( t1 )
    Inicializar r
    foreach t2 of D2(s1) do
        r ← f2( r, t2 )
    agregar s1‖r a R
```

```
R ← ∅
foreach t1 of D1 do
    Inicializar r
    foreach t2 of D2(t1) do
        r ← f2( r, t2 )
    s1 ← f1( t1, r )
    agregar s1‖r a R
```

Si se realiza antes (esquema de la izquierda) llegamos a un resultado similar al alcanzado precedentemente, existen otros campos ya calculados al cual se anexa. Si se realiza después puede llevar a múltiples citas de un mismo cómputo lo cual el optimizador lo evitará con una subconsulta

```
SELECT s1, F2( t2 )
FROM D1 JOIN D2 ON …f1( t1 )…
GROUP BY s1
```

```
SELECT f1( t1, r ), r
FROM ( SELECT D1.*, F2( t1 ) AS
r
        FROM D1 JOIN D2 ON …
        GROUP BY t1 )
```

4.5.3 Proceso algorítmico dentro de otro horizontal

Nuevamente nos encontraremos que no hay forma de expresar la colaboración de cada proceso algorítmico. Diagnosticado con mayor precisión podemos afirmar que SQL permite construir tablas de complejidad arbitrariamente grande mientras sepamos explicar como construirla registro a registro, pues eso es lo que figura dentro de un SELECT. Todos los casos en que una tabla se construye por unión de otras en número variable el lenguaje sólo puede expresarlo por un barrido con cursores incorporando los aportes de una tabla por vez. Tablas construídas mediante algoritmos no secuenciales sobre los datos se comportan como tablas almacenadas en el sistema.

Con este diagnóstico global dejamos de incluir los procesos algorítmicos dentro de este análisis.

4.5.4 Proceso horizontal dentro de proceso vertical

```
Inicializar r
foreach t1 of D1 do
    R ← ∅
    foreach t2 of D2(t1) do
        s ← f( t1, t2 )
        Grabar s‖t1‖t2 en R
    Aqui debo indicar como cada R incide
en r
```

Y nos encontramos con el problema propio de todos los anidamientos de procesos horizontales en otros. Si tuplas de todas las R inciden del mismo modo no hubiera sido necesario la subdivisión y se trata de un único proceso vertical sobre una tabla ensamblada, si las tuplas de un mismo R inciden de un mismo modo, se puede tratar como un proceso vertical dentro de otro vertical, o simplemente expresarlo en la fórmula con la cual la t actualizan la r. si todo R tiene una única

incidencia en r entonces seguro que son procesos verticales dentro de otro vertical. Estos los veremos en el punto siguiente.

4.5.5 Proceso vertical dentro de procesos verticales

```
Inicializar r
foreach t1 of D1 do
    Inicializar s
    foreach t2 of D2(t1) do
        s ← f1( s, t1, t2 )
    r ← f2( r, s, t1 )
```

Esta combinación se puede traducir con una subconsulta que fabrique los pares s, t_1 que sirven para evaluar r.

```
SELECT  F2( s, t )
FROM  ( SELECT F1( D1.t1, D2.t2 ) s, D2.t1 .t
        FROM D1 JOIN  D2 ON ... D1.t1 ...
        GROUP BY D1.t1 )
```

Este esquema de reducción se puede utilizar para procesos verticales más anidados. Las sucesivas subconsultas anidadas tendrán cada vez más ensambles y listas de agrupamiento más extensas.

4.5.6 Procesos condicionados por el resultado de otros

Aunque dentro de un proceso elemental no hay sentencias condicionales, es factible que un proceso completo quede subordinado al resultado de otro. Esto es de particular interés cuando el subordinante y el subordinados son interiores a un ciclo tanto puro como gobernando un proceso. A su vez puede manejarse la condición por un valor, fruto de un cálculo vertical como por una tabla, fruto de un cálculo horizontal. Los procesos algorítmico pueden producir tanto uno u otro.

Empezando por las condiciones gobernadas por un valor:

```
R ← ∅
foreach t1 of D1 do
    Inicializar r1
    foreach t2 of D2( t1 ) do
        r1 ← f1( r1, t1, t2 )
    if  φ( r1 )
        Inicializar r2
        foreach t3 of D3( t1 , r1 )
do
            r2 ← f2( r2, t1, t3, r1 )
        agregar t1‖r1‖r2 en R
```

Por supuesto que muchos parámetros que hemos agregado como posibles en los casos concretos no aparecen.

Un modo sistemático de reducirlo es calcular primero una relación de los pares (t_1 , r_1) para los cuales debe realizarse el segundo cálculo y usar este como tabla de entrada del segundo proceso.

```
foreach t1 of D1 do
    Inicializar r1
    foreach t2 of D2( t1 ) do
        r1 ← f1( r1, t1, t2 )
    if  φ( r1 )
        grabar t1‖r1 en R1
R ← ∅
foreach t1‖r1 of R1 do
    Inicializar r2
    foreach t3 of D3( t1 , r1 ) do
        r2 ← f2( r2, t1, t3, r1 )
```

```
agregar t1‖r1‖r2 en R
```

En SQL es muy sencillo arrancar un cálculo a partir de una resultado intermedio, basta con escribir a éste como una subconsulta en el FROM

```
SELECT aux.*, F2( aux.*, D3.t3.)
FROM ( SELECT D1.*, F1( D1.t1, D2.t2 )
       FROM D1 JOIN D2 ON ... .D1.t1 ...
       GROUP BY D1.t1
       HAVING φ( r1 )  ) aux  JOIN D3 ON  aux.t1  aux.r
```

Si la condición es gobernada por una tabla, cabe preguntarse que se puede preguntar sobre ella, cualquier propiedad que se traduzca en un sí o no puede transformarse en un valor (proceso vertical) que lo represente. Así preguntar si una tabla es vacía equivale a preguntar si la cardinalidad es 0. Sin embargo como la pregunta por vacío existe en SQL bajo la modalidad EXIST podemos analizarla un momento. Otras preguntas vinculadas a la naturaleza de conjunto que tienen las tablas se pueden reducir a esta pregunta, así preguntar si una está contenida en otra equivale a preguntar si la resta es vacía. Veamos un esquema genérico basado en conjunto vacío

```
…
foreach t of D1 do
    Proceso horizontal H que
determi
     na un R a partir de un D2( t
)
    if  R > ∅
        un proceso P contribuye al
          resultado buscado
```

Determinamos los valores de t1 para los cuales se quiere realizar el segundo proceso, seguido de otro ciclo donde efectivamente se computan

```
aux ← ∅
foreach t of D1 do
    Proceso H que determina un R a partir de un D2( t )
    if  R > ∅
        grabar t en aux
…
foreach t of aux do
    un proceso P contribuye al resultado buscado
```

El ensamble de D_1 y el proceso que construye R no contiene registros si R es vacío, tomando su proyección sobre los elementos de D_1 se tiene un resultado que permite saber con un IN si un cierto t debe ser procesado o no en la segunda parte. El proceso H no aporta nada al resultado ya que en la práctica no recibe una evaluación completa.

```
SELECT ...
FROM  D1
WHERE  t  IN  ( SELECT ... )
```

Si la pregunta fuera por vacío se puede negar la condición "IN ..."

4.5.7 Múltiples procesos dentro de otro

Esta situación se puede equiparar a lo estudiado en *4.4.6* considerando que el primer proceso tomó la decisión de que todos los argumentos eran válidos. Observen que con la máxima generalidad consideramos que el resultado del primer proceso no sólo había servido para tomar la decisión sino que estaba a disposición del segundo proceso.

4.5.8 Reglas generales de reducción

Para documentar mejor estos esquemas abstractos hemos elaborado ejemplos concretos.[2] De este análisis exhaustivo de casos podemos establecer una estrategia general de reducción de procesos anidados

1) Secuencias de procesos dentro de un ciclo, sean de ejecución condicionada o no, se resuelven por una secuencia de ciclos con el mismo control ejecutando un solo proceso interior por vez.
2) Múltiples anidamientos de procesos horizontales, que contribuyen a una única relación objetivo, se deben encarar por unión de múltiples procesos con un solo proceso horizontal anidado.
3) Un proceso horizontal anidado pierde su anidamiento si se lo programa sobre un ensamble.
4) Un proceso vertical anidado se elimina mediante una subconsulta.
5) Una pregunta que condiciona una elaboración al resultado de otra se resuelve mediante una subdivisión del proceso en dos, en una tabla auxiliar se elaboran los elementos habilitados para el segundo proceso. Finalmente ambos se pueden unir sintácicamente recurriendo a la construcción `IN` evitando el agregar momentáneamente una tabla auxiliar al catálogo.

5. CUANDO EL USO DE CURSORES ES INEVITABLE

Hemos detectado un solo caso en el cual los cursores son inevitables y ello corresponde a una situación completamente distinta a la analizada. Nuestro análisis estudió las elaboraciones que comienzan con una vista, armada tal vez con ayuda de varias tablas, a partir de la cual se quiere computar una nueva tabla.

En toda vista la cantidad de tablas que intervienen es perfectamente conocida por quien la escribe. ¿Qué pasa si las tablas sobre las cuales quiero actuar las describo por alguna característica? Podría no saber a priori, o no querer contabilizarlo, cuantas aparecerán. Esto es factible recurriendo a la meta-información almacenada en la misma base.

Obsérvese que el argumento de un `FROM` es un conjunto concreto de tablas, aún recurriendo a subconsultas. Se puede ampliar la cantidad pero quién la escribe sabe a cuantas tablas afecta su consulta. También la `UNION` cita una cantidad concreta. Podríamos decir que las tablas intervienen como constantes. No creemos que tenga sentido mantener el recurso cursores para esta única finalidad. Los ejemplos que hemos encontrado son acciones de unión entre contenidos de campos homónimos de muchas tablas y sólo durante tareas de remodelación de bases de datos. Aún usado en esta situación los cursores dispararían la acción sobre cada tabla en secuencia frenando una posible ejecución paralela si las tablas involucradas residieran en más de un procesador.

6. RESUMEN, CONCLUSIONES Y EXTENSIONES

A lo largo de este análisis hemos señalado los aspectos en los cuales debiera ser enriquecido el SQL92, punto de referencia para muchos motores. Estos son:

- Proveer funciones sin ninguna limitación algorítmica. Éstas en tres en modalidades, todas ellas capaces de entregar tuplas con varios atributos.
    - A partir de los campos de una tupla para elaborar nuevos campos de esa tupla.
    - A partir de una o más columnas de una tabla elaborar nuevos campos con funciones simétricas.
    - A partir de una tabla engendrar otra tabla o una tupla en forma algorímica.
- Extender el segundo mecanismo detallado en el punto anterior a operadores solo asociativos a ser utilizados en contextos de ORDER BY.
- Agregar un mecanismo para uniones genéricas de tablas.

Todas las características enunciadas aumentarían simultáneamente la expresividad del lenguaje y el paralelismo en la ejecución y permitirían conservar la característica descriptiva y funcional para

cada consulta. Una vez hecha esas mejoras el uso de cursores, que destruye las características de lenguaje, se vuelve superfluo.

A partir de este trabajo hemos abierto varias líneas de trabajo adicionales:

- Este trabajo debiera ser extendido para empalmar con versiones más modernas de SQL (1999 y 2003), aún no plena y masivamente implantadas.
- También se intentará concretar una propuesta concreta de modificación del lenguaje analizando las modificaciones gramaticales necesarias.
- La experiencia docente sugiere que la redacción de sentencias SQL complejas no resulta intuitiva. De los esquemas de reducción discutidas surge la posibilidad de ofrecer un lenguaje alternativo que permita describir con mentalidad imperativa lo deseado y lo traduzca al SQL facilitando el uso de los motores existentes.
- En varios casos hemos encontrado formas alternativas de expresar un requerimiento en SQL, no así en el sentido inverso, donde la expresión imperativa admite menos alternativas. Esto sugiere que alrededor de ella se podría tener una forma normal interna, punto de partida para encarar optimizaciones en la evaluación, de modo tal que el estilo de redacción de la consulta no influyera en la eficiencia de la misma.

AGRADECIMIENTOS

Al Ing. Jorge Doorn por haber leído y hechos valiosas sugerencias para este artículo.

REFERENCIAS

[1] Chigrik, Alexander "*Using SQL Server Cursors"* www.mssqlcity.com/Articles/General/UseCursor.htm

[2] Grupo Cursores "*Informe técnico N°2*" del en Depto. Ingeniería Univ. Nac. de la Matanza.

[3] Melton Jim (ed.) "*(ISO-ANSI Working Draft) Foundation*", August 2003, pág. 505

[4] Thayer, Matías "*Cursores en SQL Server*" http://maestrosdelweb.com/editorial/cursql

Desarrollo de un Método de Evaluación de la Madurez para Pequeñas y Medianas Empresas de Software[1]

Alicia Mon, Marcelo Estayno, Andrea Arancio
{aliciamon; mestayno; andrea.arancio@fibertel.com.ar}
G.I.S.[2] - UNLaM
Argentina

Abstract

The industry of software has become an economic activity of extreme importance of the Latin American countries, where the small and medium companies (PyMES) of software development have marked to a strong presence, conforming an integrated and central part of the productive organizations.

This productive impulse imposes some modifications in the work processes and leads to the PyMES to a process of improvement and certification of quality to position itself competitively in the national and international market. However the processes improvement, in these organizations, still is in an immature state.

The current models of Process there are in the market, such as CMMI and/or the Norms of Quality ISO are complex in their implementation, difficult to fulfill and of high cost for the PyMES of the sector.

The MoProSoft Model of Process [1] has been created in Mexico like proposal to solve those problems of the small organizations and oriented to improve its processes.

The present article exposes the results of the development and the application of a method of evaluation based on MoProSoft Model, which allows to verify if the model before mentioned is really effective and to detect the necessities of improvement in the construction of a Latin American model.

Keywords: Models of Process. Software Quality. PYMEs.

Resumen

La industria del software se ha convertido en una actividad económica de suma importancia para la mayoría de los países latinoamericanos, donde las pequeñas y medianas empresas (PyMES) de desarrollo han marcado una fuerte presencia, conformando una parte integrada y central de las organizaciones productivas.

1 El presente trabajo está enmarcado dentro del proyecto 506AC0287-COMPETISOFT - Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria de Software de Ibero América - del programa CYTED (Ciencia y Tecnología para el Desarrollo) .

2 GIS - Grupo de Ingeniería de Software. Secretaría de Posgrado. Departamento de Ingeniería e Investigaciones Tecnológicas. Universidad Nacional de La Matanza.

Este impulso productivo, impone algunas modificaciones en los procesos de trabajo y conduce a las PyMES a un proceso de mejora y certificación de calidad para posicionarse competitivamente en el mercado nacional e internacional. No obstante la mejora de los procesos en estas organizaciones todavía se encuentra en un estado inmaduro.

Los modelos de Proceso existentes en el mercado, tales como CMMI y/o las Normas de Calidad ISO resultan complejos en su implementación, difíciles de cumplir y de alto costo para las PyMES del sector.

El Modelo de Proceso MoProSoft [1] ha sido creado en México como propuesta para resolver esos problemas de las pequeñas organizaciones y orientado a mejorar sus procesos.

El presente artículo, expone los resultados del desarrollo y la aplicación de un método de evaluación basado en el Modelo MoProSoft, que permita comprobar si el modelo antes mencionado es realmente eficaz y detectar las necesidades de mejora en la construcción de un modelo iberoamericano.

Palabras Clave: Modelos de Proceso. Calidad de Software. PYMEs.

1. INTRODUCCIÓN

El desarrollo tecnológico y la competitividad del mercado mundial de la primera década del siglo XXI, indica que producir software sin la correspondiente calidad asociada redunda en proyectos destinados al fracaso [2]. El desarrollo de software con niveles aceptables de calidad se presenta como una necesidad que debe ser tenida en cuenta tanto por las empresas desarrolladoras de software como por los clientes y usuarios de los mismos.

El papel de las pequeñas y medianas empresas (PyMES) en la estructura industrial se ha conformado como una parte integrada y no simplemente alternativa de organizaciones productivas. Especialmente la industria del software, se ha convertido en una actividad económica de suma importancia para la mayoría de los países latinoamericanos, donde las Pymes han marcado una fuerte presencia.

En este contexto, la calidad del software es un concepto complejo que se presenta "por comparación" entre productos desarrollados y su especificación [3]. La calidad del producto está estrechamente vinculada a la calidad del proceso de desarrollo. Numerosos Estándares de Proceso [4], [5], proponen ordenar en forma prescriptiva las actividades que deben realizarse a través del desarrollo [6].

Las pequeñas y medianas empresas desarrolladoras de software sufren el cambio de paradigma, producto de la maduración del mercado del software, y es que ya no basta con aplicar bien la tecnología, o aplicar la tecnología de última generación para obtener un buen producto software, sino que la única forma que tiene una Pyme de desarrollo de Software de mejorar su eficiencia y ser productiva alcanzando los niveles de calidad exigidos por el comercio exterior, es incorporando un modelo de calidad, que se ajuste a las necesidades del tipo de organización.

Los diferentes Modelos de Proceso Software y Normas de Calidad existentes en el mercado para medir o certificar los procesos de desarrollo (ISO/IEC 15504-2, ISO 90003, ISO 9001:2000, CMMI) [7], [8], [9] [10], incluyen conjuntos de procesos y actividades que responden a los criterios de desarrollo de sistemas complejos, de tipos de organizaciones grandes y con estructuras formales muy definidas. pero los mismos son tan complejos en su implementación para las, que las mismas Para este tipo de empresas debe adoptarse algún tipo de práctica que permita trabajar con normas de calidad adaptadas a su entorno [11].

En este sentido, las PyMES de la industria del software poseen un conjunto de características que presenta serias dificultades a la hora de adecuar sus actividades para conducir a sus organizaciones en un proceso de certificación con estos Modelos, dado que las pocas facilidades de financiación, los problemas para planear su crecimiento, la falta de gerenciamiento profesional, las dificultades para exportar, y los Sistemas de información, administración y contabilidad deficientes, son expresiones claras de las limitaciones del sector para implementar dichos modelos.

No obstante, este tipo de empresas del sector, poseen una serie de ventajas que les permitiría mejorar su competitividad, dado que poseen mayor posibilidad de flexibilidad y de reacción frente a los cambios, mayor poder de innovación, menores costos de infraestructura, puntos de ventas cercanos al consumidor, atención más directa y personalizada con los clientes.

En este sentido, resulta necesario desarrollar un modelo de procesos que asegure obtener los niveles de calidad requeridos por el mercado, pero que resulte sencillo de implementar y fácil de entender por los equipos de desarrollo de las PyMES, y que pueda incorporarse en forma gradual para que las empresas puedas percibir los logros sin detener su producción.

El Modelo de Procesos Mexicano MoProSoft [1] ha sido elaborado como un modelo de madurez, con diferentes niveles, adaptado a las necesidades de las pequeñas y medias empresas en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua.

El proyecto COMPETISOFT [12] en el cual se enmarca la presente investigación, tiene por objetivo incrementar el nivel de competitividad de las PyMES Iberoamericanas productoras de software mediante la creación y difusión de un marco metodológico común que pueda llegar a ser la base sobre la que se establezca un mecanismo de evaluación y certificación de la industria del software reconocido en toda Ibero América, tomando como base los diferentes modelos existentes, especialmente la experiencia mexicana.

En el presente artículo, se expone el desarrollo de un cuestionario que conforma un Método de medición y evaluación de uno de los procesos del Modelo MoProSoft, así como los resultados obtenidos de su aplicación en un conjunto seleccionado de 15 (quince) empresas de desarrollo de software.

2. DESARROLLO DE UN MÉTODO DE EVALUACIÓN

El Modelo MoProSoft está estructurado en base a 3 (tres) categorías que abarcan las responsabilidades asociadas con la organización: Alta Dirección, Gerencia y Operación. Dentro de cada categoría se definen un conjunto de procesos que incluyen prácticas y roles específicos.

La categoría 1- Alta Dirección, aborda las prácticas relacionadas con la Gestión del Negocio, proporciona los lineamientos a los procesos de la categoría de Gerencia y se retroalimenta con la información generada por ellos. La categoría 2-Gerencia, aborda las prácticas de Gestión de Procesos, Proyectos y Recursos en función de los lineamientos establecidos en el nivel de Alta Dirección y proporciona los elementos para el funcionamiento de los procesos de la categoría de Operación, recibe y evalúa la información generada por éstos y comunica los resultados a la Alta Dirección.

La categoría 3- Operación se subdivide en dos procesos, Administración de Proyectos Específicos y Desarrollo y Mantenimiento de software. El primer proceso busca establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados. El segundo, apunta a la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos o modificados cumpliendo con los requerimientos especificados. En este nivel se realizan las actividades de acuerdo a los elementos proporcionados por el nivel de Gerencia y entrega a ésta la información y productos generados.

Estas tres categorías se encuentran relacionadas entre sí a través de los diferentes procesos y los productos de entrada que cada uno requiere y de salida que cada uno genera.

La siguiente Figura 1 presenta el esquema del conjunto de procesos por categoría que define el modelo MoProSoft:

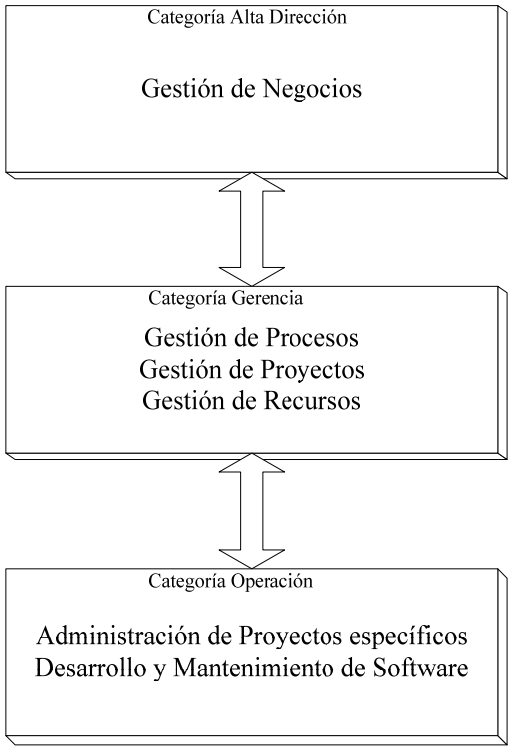


Figura 1. Estructura del Modelo MoProSoft

Para alcanzar diferentes niveles de madurez, MoProSoft propone un esquema de seis niveles de capacidad, alcanzables por una empresa de desarrollo de software: 0-sin proceso definido, 1-Realizado, 2-Gestionado, 3-Establecido, 4-Predecible y 5-Optimizado. El Modelo identifica cada nivel de capacidad con un color diferente. Los colores sugieren un ordenamiento de la implementación de las prácticas de los procesos de MoProSoft, partiendo de las actividades básicas, correspondientes al nivel 1-Realizado, e incorporando sucesivas prácticas que corresponden al resto de los niveles más avanzados.

La siguiente Tabla 1 refleja la correspondencia entre los niveles de capacidad de procesos y los colores que los representan.

Tabla 1. Niveles de madurez del Modelo MoProSoft

Nivel	Capacidad de proceso	Color
1	Realizado	Amarillo
2	Gestionado	Azul
3	Establecido	Verde
4	Predecible	Rosa
5	Optimizado	Ninguno

El método de evaluación desarrollado, consiste en un cuestionario [13] orientado al Proceso de Administración de Proyectos Específicos de MoProSoft, que tiene por objetivo realizar una evaluación sobre el cumplimiento de las prácticas definidas por el Modelo MoProSoft para este proceso y establecer el nivel en que se encuentra una empresa que comienza a implementar este Modelo de Madurez, así como establecer los procesos de mejora necesarios [14].

Las preguntas del cuestionario han sido desarrolladas en base a las actividades planteadas en este proceso específico, el que se divide en 4 fases: A1.- Planificación, A2.-Realización, A3.-Evaluación y Control y A4.-Cierre.

El conjunto de preguntas respeta los diferentes niveles de madurez definidos para cada una de las fases y tiene un total de 319 preguntas divididas en las 4 Fases, con la distribución que se presenta en la Tabla 2.

Tabla 2. Cantidad de preguntas por fase en el cuestionario

Fases	Cantidad de preguntas
Planificación	146
Realización	97
Evaluación y Control	44
Cierre	32

El cuestionario abarca, para cada práctica, las instancias básicas que deben ser cubiertas. Cada práctica definida en el Modelo se integra con un conjunto de preguntas que procura identificar cuanto y como se realiza dicha práctica. Asimismo, cada pregunta tiene un nivel de madurez, asociado al nivel de madurez del Modelo, que se representa en forma coloreada.

Las preguntas se correlacionan con los niveles de capacidad de procesos de la siguiente forma, tal como se muestra en la Tabla 3.

Tabla 3. Cantidad de preguntas por nivel de madurez en el cuestionario

Nivel	Capacidad de proceso	Cantidad de preguntas
1	Realizado	66
2	Gestionado	105
3	Establecido	139
4	Predecible	9
5	Optimizado	

Cada pregunta puede tener diferentes tipos de respuestas: Si/No, Roles (de acuerdo a los Roles definidos en el Modelo), opciones múltiples o texto libre, para aquellas respuestas que no son conducidas y pueden contemplar diversas opciones. El encadenamiento propuesto entre las preguntas se ha establecido en función de las respuestas obtenidas.

3. APLICACIÓN DEL MÉTODO DE EVALUACIÓN

El cuestionario ha sido aplicado, a modo de prueba, por el G.I.S. [15] en 15 (quince) empresas y grupos de desarrollo diferentes, dentro de los cuales se encuentran: 4 (cuatro) empresas grandes, 5 (cinco) empresas medianas, 5 (cinco) PyMEs y un Organismo Público, lo que ha permitido lograr una base de datos que permite analizar de forma más detallada el Modelo MoProSoft y el cuestionario de evaluación.

Las empresas en las que se aplicó el cuestionario han seleccionado un Proyecto específico, conformado por un pequeño equipo de desarrollo, para poder medir la ejecución de las prácticas que define el Modelo. Las características principales de los proyectos en los que se ha probado el cuestionario, como herramienta de evaluación, se presentan en la siguiente Tabla:

Tabla 4. Características de las empresas y los proyectos específicos

Tamaño Empresa / Organización	Tipo de Empresa /Actividad	Personas asignadas al proyecto
Grande	Empresa de Desarrollo de Software	21
Grande	Telecomunicaciones	4
Grande	Empresa de Desarrollo de Software	12
Grande	Multimedia	5
Mediana	Consultora de Desarrollo de Software	5
Mediana	Empresa de Desarrollo de Software	16
Mediana	Droguería	5
Mediana	Empresa de Desarrollo de Software	8
Mediana	Consultora de Desarrollo de Software	3
Pequeña	Consultora de Desarrollo de Software	5
Pequeña	Consultora de Desarrollo de Software	6
Pequeña	Empresa de Servicio	4
Pequeña	Consultora de Desarrollo de Software	7
Pequeña	Consultora de Desarrollo de Software	2
Grande	Organismo Público	4

Los resultados de la aplicación del cuestionario, han permitido realizar un análisis cuantitativo y un análisis cualitativo sobre las actividades del Proceso de Administración de Proyectos Específicos, que realizan las empresas evaluadas, así como analizar cual es el porcentaje de actividades adecuado para que una PyME realice y pueda mejorar sus procesos.

La Tabla 5, que se presentan a continuación, expone los datos obtenidos en cada uno de los proyectos representados, de acuerdo a las siguiente información: Tamaño de Empresa; Tipo de Empresa; Cantidad de personas asignadas a cada proyecto (PAP); porcentaje de respuestas positivas obtenidas en cada uno de los niveles de madurez (1-Realizado,2-Gestionado, etc,) y la columna "Total" presenta el porcentaje de respuestas positivas obtenidas sobre el total de las 319 preguntas del cuestionario. La tabla se ha ordenado por tamaño de empresas y a su vez por el porcentaje total de respuestas obtenidas. El nivel 5 – Optimizado no se encuentra presente debido a que en el Modelo no existen actividades del mayor nivel de madurez en este proceso analizado.

Tabla 5. Porcentaje de respuestas obtenidas de los Proyectos, divido por niveles de Madurez de MoProSoft

Tamaño	Tipo de Empresa	PAP	1 – Realizado	2 – Gestionado	3 – Establecido	4 – Predecible	Total
Grande	Empresa de Desarrollo de Software	21	85%	83%	64%	78%	75%

Grande	Telecomunicaciones	4	68%	80%	43%	22%	60%
Grande	Empresa de Desarrollo de Software	12	70%	66%	38%	22%	53%
Grande	Multimedio	5	77%	49%	32%	44%	47%
Mediana	Empresa de Desarrollo de Software	8	91%	60%	54%	78%	64%
Mediana	Consultora de Desarrollo de Software	5	82%	69%	48%	67%	62%
Mediana	Empresa de Desarrollo de Software	16	67%	57%	53%	22%	56%
Mediana	Consultora de Desarrollo de Software	3	80%	53%	37%	67%	52%
Mediana	Droguería	5	82%	53%	33%	22%	50%
Pequeña	Consultora de Desarrollo de Software	7	82%	82%	52%	78%	69%
Pequeña	Consultora de Desarrollo de Software	2	70%	82%	58%	67%	68%
Pequeña	Consultora de Desarrollo de Software	5	85%	69%	46%	100%	63%
Pequeña	Consultora de Desarrollo de Software	6	65%	58%	29%	22%	46%
Pequeña	Empresa de Servicio	4	64%	37%	17%	44%	34%
Grande	Organismo Público	4	50%	31%	15%	0%	27%

A partir de la aplicación del cuestionario y el análisis cuantitativo presentado en la tabla 5, se ha realizado un análisis comparativo entre los resultados obtenidos de las diferentes empresas y los porcentajes de respuestas obtenidas para cada nivel, tal como presenta el Gráfico 1.

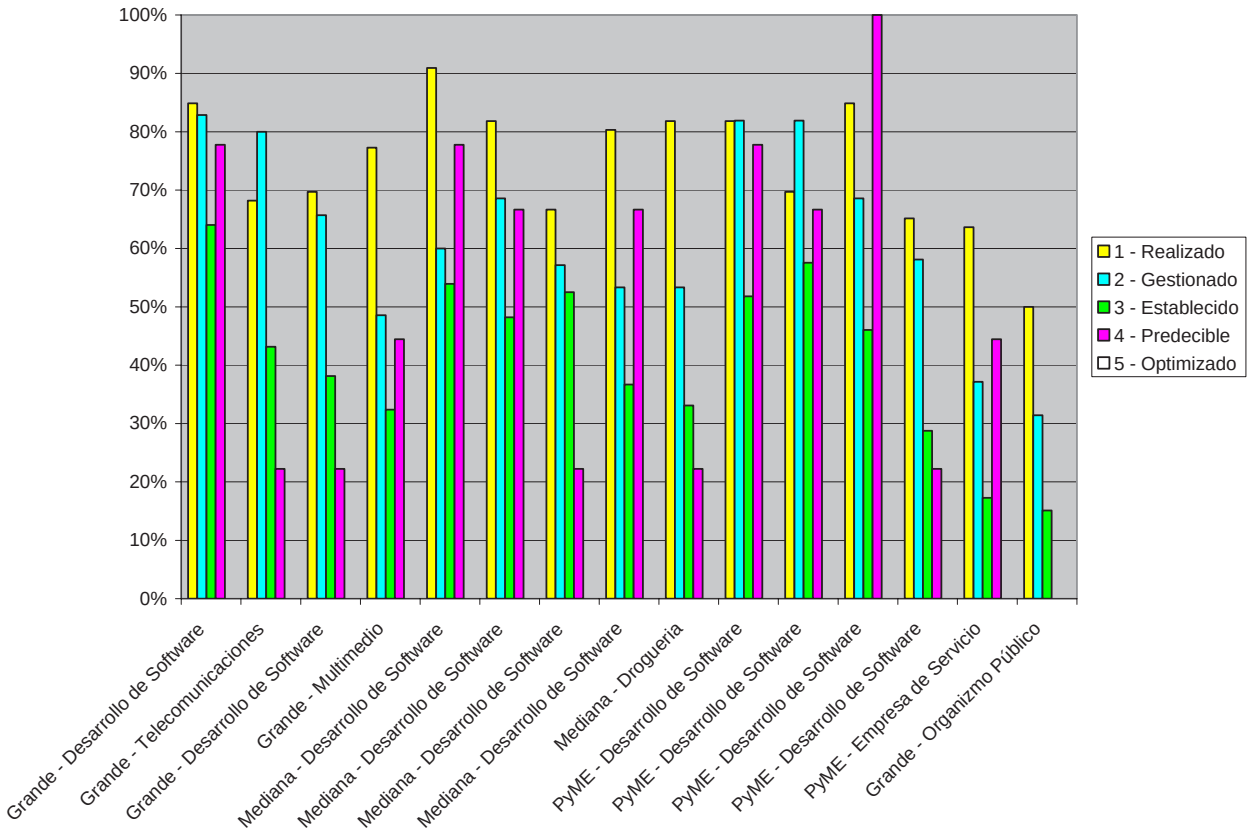


Gráfico 1. Comparación entre las empresas a través de los porcentajes de las respuestas obtenidas por cada nivel

Este gráfico permite observar los porcentajes de realización de cada nivel de madurez, en el Proceso de Administración de Proyectos Específicos, por cada una de las empresas evaluadas, expresado en los colores correspondientes:

Porcentaje de preguntas positivas en el Nivel 1 – Realizado

Porcentaje de preguntas positivas en el Nivel 2 – Gestionado

Porcentaje de preguntas positivas en el Nivel 3 – Establecido

Porcentaje de preguntas positivas en el Nivel 4 – Predecible

Por otra parte, la siguiente Tabla 6, muestra el resultado promedio obtenido de la cantidad de respuestas positivas obtenidas por la totalidad de las 15 empresas evaluadas en cada uno de los niveles de madurez del Proceso.

Tabla 6. Porcentaje de respuestas obtenidas en cada nivel de Madurez de MoProSoft

Nivel	Capacidad de proceso	Porcentaje
1	Realizado	74%
2	Gestionado	62%
3	Establecido	41%
4	Predecible	49%
5	Optimizado	

4. ANALISIS DE RESULTADOS

Del análisis de los resultados de la aplicación del cuestionario en las empresas seleccionadas, se detecta que, comenzando con el análisis del nivel 1 de Madurez (Realizado - color amarillo), en promedio cumplen con un 74% en la realización de las Actividades especificadas por el Proceso de Administración de Proyectos Específicos, así como con la documentación que las mismas requieren para su sistematización.

De las Actividades del Proceso definidas para el nivel 2 – Gestionado (Azul), el promedio extraído de la capacidad de dicho nivel es del 62%. No obstante, una parte de estas actividades no quedan documentadas debidamente.

Si bien estos promedios indican que no existe un salto cuantitativamente significativo entre la Realización (74%) y la Gestión (62%) de las actividades definidas por el Modelo, existe una diferencia importante a la hora de analizar cada uno de los casos dependiendo del tipo de organización de que se trate. Por otra parte, en general se evidencia una notable diferencia entre los niveles de madurez mas elevados como son los niveles 3-Establecido (verde) con un promedio del 41% y del Nivel 4-Predecible (rosa) con un 49% en la realización de las actividades del Proceso.

En el análisis integral del Modelo, con los diferentes procesos vinculados, se detecta que las actividades que no se realizan en los niveles 1 y 2 están estrechamente vinculadas con actividades definidas para otros Procesos de MoProSoft, específicamente de la categoría Gerencia, como son los Proceso de Gestión de Recursos y/o Gestión de Procesos y de la categoría de Alta Dirección, que involucra el Proceso de Gestión de Negocios. Esto indica que las empresas evaluadas requieren un proceso de mejora en las actividades que se encuentran definidas en dichos Procesos y es allí donde debiera focalizarse un Plan de Mejora a implementar para cada una de ellas.

La Administración de Proyectos Específicos en la mayoría de las empresas evaluadas, se realiza en un gran porcentaje (74%) y las actividades que propone el Modelo de Proceso han sido verificables con facilidad aplicando el método de evaluación desarrollado, con el conjunto de preguntas y registros que permite el cuestionario.

5. CONCLUSIONES y TRABAJOS FUTUROS

El cuestionario desarrollado ha permitido evaluar todas las actividades del Proceso de Administración de Proyectos Específicos del Modelo MoProSoft, posibilitando el análisis en los casos de prueba de la realización, registro y documentación de las mismas. Así, el cuestionario logró pasar test de aceptación en quince (15) contextos diferentes, y permitió establecer la consistencia del mismo.

Con los resultados de la prueba, quedan por establecer patrones de análisis para la elaboración de un Método de Evaluación que integre todo el Modelo de Proceso y que defina los porcentajes mínimos necesarios de realización de cada actividad. Por otra parte, se considera que con el Modelo aplicado y el método que se encuentra en construcción, es posible identificar en las empresas evaluadas sus conductas de calidad en lo que hace a la Administración de Proyectos Específicos.

Como trabajo a futuro, se está avanzando en dos aspectos. Por una parte, en la ampliación del cuestionario de evaluación a otros procesos y por otra parte en la validez y sencillez del Modelo MoProSoft para ser aplicado en pequeñas y medianas empresas de desarrollo de software. Finalmente.

El primer aspecto incumbe a la ampliación del método de evaluación con el desarrollo de otros cuestionarios que aborden los diferentes procesos de la estructura del modelo MoProSoft. La estrategia de ampliación del método continúa por el Proceso de Desarrollo y Mantenimiento, para poder evaluar integralmente todo el Proceso de Operación y en la definición de los porcentajes mínimos de realización que permita definir un parámetro de realización y una estrategia de mejora en el proceso.

El segundo aspecto, se refiere a la construcción de un Modelo de Madurez sencillo de aplicar, entendible en su estructura y lenguaje, y adecuado a las PyMES Iberoamericanas, tomando como base el Modelo MoProSoft, con modificaciones, mejoras y/o agregados basados en los diferentes procesos analizados y probados.

BIBLIOGRAFÍA

[1]. Oktaba, H; Esquivel, C; et al. Modelo de Procesos para la Industria del Software. MoProSoft. Versión 1.3. Mayo 2005.

[2]. Pflegger, S: Ingeniería de Software. Teoría y Práctica. Prentice Hall. (2002)

[3]. Crosby, P: Quality Is Free. New York: McGraw-Hill. (1979)

[4]. IEEE Standard 1074-1997, IEEE Standard for Developing Software Life Cycle Processes,. IEEE, (1997).

[5]. ISO/IEC. International Standard: Information Technology. Software Life Cycle Processes, ISO/IEC Standard 12207-1995/Amd. 1-2002.

[6]. S.Acuña, N.Juristo A.Moreno,A.Mon: A Software Process Model Handbook for Incorporating People's Capabilities.. Springer; 1 Edition, (2005)

[7]. ISO/IEC. ISO/IEC TR 15504. Information Technology – Software process assessment. International Organization for Standardization, International Electrotechnical Commission, 1998. http://www.sel.iit.nrc.ca/spice

[8]. ISO/IEC 9000-3:1997. Quality management and quality assurance standards. Part 3: Guidelines for the application of ISO 9001: 1994 to the development, supply, installation and maintenance of computer software. International Organisation for Standardization, ISO, 1997.

[9]. Capability Maturity Model® Integration (CMMISM), Version 1.1. CMMISM for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing. (CMMI-SE/SW/IPPD/SS, V1.1). Software Engineering Institute, Carnegie Mellon University. 2002.

[10]. Ahern, D.M; Clouse, A., et al. CMMI Distiller. Reading, MA: Addison-Wesley. (2001)

[11]. Pino, F.; García, F; Piattini, M.: Adaptación de las normas ISO/IEC 12207:2002 e ISO/IEC 15504: 2003 para la evaluación de la madurez de procesos de software en países en desarrollo.

[12]. Competisoft. Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica. Versión 0.2. Diciembre 2006. Proyecto COMPETISOFT 506PI287. CYTED- Programa Ibero-Americano de Ciencia y Tecnología para el Desarrollo. http://www.cyted.org

[13]. Bertone, R; Pasini, A; Ramon, H, et al. Gestión de Calidad en la Construcción del Software. Un enfoque para PyME's. Anales CACIC 2006. San Luis. Argentina. (2006)

[14]. Estayno, M.; Mon, A; De Maria, E; Arancio, A, et al. Cuestionario para la evaluación de PyMEs desarrolladoras de Software. Administración de Proyectos Específicos. Informe Técnico Comisión de Investigaciones Científicas de la Provincia de Buenos Aires. GIS Departamento de Ingeniería. Universidad Nacional de La Matanza. LIDI. Facultad de Informática. Universidad Nacional de La Plata. Agosto 2006. CIC.

[15]. Grupo de Ingeniería de Software. Departamento de Ingeniería e Investigaciones Tecnológicas. Escuela de Posgrado. Universidad Nacional de La Matanza.

Developing a Repository of Knowledge for Virtual Communities with Semantic Web Technologies

M. Clara Casalini, Elsa Estevez and Pablo Fillottrani
Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur
Bahía Blanca, 8000, Buenos Aires, Argentina
{mcca, ece, prf}@cs.uns.edu.ar

Abstract

The appearance and continuous growth of virtual communities on the web imposes the challenge of coping with the large amount of information and knowledge these communities produce. The Semantic Web with its set of standards and technologies provides the basic means for implementing repositories of knowledge for virtual communities. However not every member of a virtual community knows these technologies and is ready to use them. We present an implementation of a repository of knowledge that follows the Resource Description Framework enabling a standard representation of knowledge in a community and providing the necessary functionality for members of the community to manage this repository introducing information and relating this information in a simple manner that also facilitates the interoperability with other repositories.

Keywords: Virtual Communities, Knowledge Management, Semantic Web, Interoperability, Web Standards

1 INTRODUCTION

A virtual community is defined in [12] as "an aggregation of individuals or business partners who interact around a shared interest, where the interaction is at least partially supported and/or mediated by technology". Whether oriented to business, entertainment or education, groups of people who gather to share and interact sometimes without even knowing each other and living very far from the rest are becoming more popular every day. Many of these communities share a set of assets which is developed by the members of the community and becomes larger and more vital for the community as it grows. The spread of virtual communities leads researchers to the task of finding the best way to represent these assets and to implement a virtual repository for their knowledge.

The Semantic Web is emerging as a second, enhanced version of the World Wide Web whose mission is not only to hold information, but to allow sharing and reusing that information, giving it a unique meaning and providing the means for automating the tasks related to processing that information. In this context, the work under development to define what the Semantic Web is and how it is going to be implemented seems like the ideal framework for the applications needed by virtual communities.

This paper presents an implementation of a repository of knowledge to support virtual communities. The underpinning technologies of its design and implementation are the technologies being developed for the Semantic Web. Section 2 presents and explains these technologies together with some examples of web sites which are already applying them. The development of the repository is presented in section 3. The complete process is explained from the requirements elicitation (section 3.1), through domain and use case modeling (sections 3.2 and 3.3), architectural and detailed design (sections 3.4 and 3.5) to the implementation (section 3.6). Section 4 is dedicated to showing

a concrete application of the system developed. An exploration of the related work and some the conclusions are given in section 5.

2 RDF AND SEMANTIC WEB

Currently, the World Wide Web is a place for sharing human-oriented information, like documents, pictures, or multimedia files. The Semantic Web [2, 4] is an evolving extension of the Web which provides a common framework that allows all kinds of data to be shared and reused across applications, enterprises and community boundaries. In this way, data would also be computer-understandable, enabling software agents to help humans with the tedious work of finding, sharing and combining information on the web, boosting B2B applications and e-commerce.

Semantic Web technologies are being developed by the World Wide Web Consortium (W3C) [1], mostly in the form of formal specifications. However, they are still very much in their infancies and their applications are limited. It comprises the standards of XML [11], XML Schema [20], RDF and RDF Schema [7, 3], and OWL [8], in a layered way (see Figure 1). XML provides an elemental syntax for content structure within documents, yet associates no semantics with the meaning of the content contained within. XML Schema is a language for restricting the structure and content of elements contained within XML documents, providing also standard datatype definitions. RDF is a simple language for expressing data models, which refers to objects (which are called resources and represented by URIs) and their relationships. An RDF-based model can be represented in XML syntax. RDF Schema is a vocabulary for describing properties and classes of RDF-based resources, with semantics for generalized-hierarchies of such properties and classes. Finally, OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. ”exactly one”), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. Thus, higher layers include languages with more expressivity but at the same time with more difficult inference procedures, belonging to harder complexity classes. The appropriate language for a given Semantic Web application is decided by making balance between being semantically more specific (and thus improving interoperability of data), and the time and space needed to evaluate the data.

RDF data model provides the elementary tools to model information, by defining ontologies which are shared conceptualizations of terms from a given domain. The RDF metadata model is based upon the idea of making statements about resources in the form of subject-predicate-object expressions, called triples in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. For example, one way to represent the notion “This document is in English” in RDF is as a triple of specially formatted strings: a subject denoting ”this document” a predicate denoting “has language”, and an object denoting “English”. Currently most known applications of RDF are RDF Site Summary [13], one of several RSS languages for publishing information about updates made to a web page, which is often used for disseminating news article summaries and sharing weblog content; FOAF (Friend of a Friend) [17] designed to describe people, their interests and interconnections; Music Brains [9] that publishes information about music albums, and SIOC (Semantically-Interlinked Online Communities) [14] designed to describe online communities and to create connections between Internet-based discussions from message boards, weblogs and mailing lists.

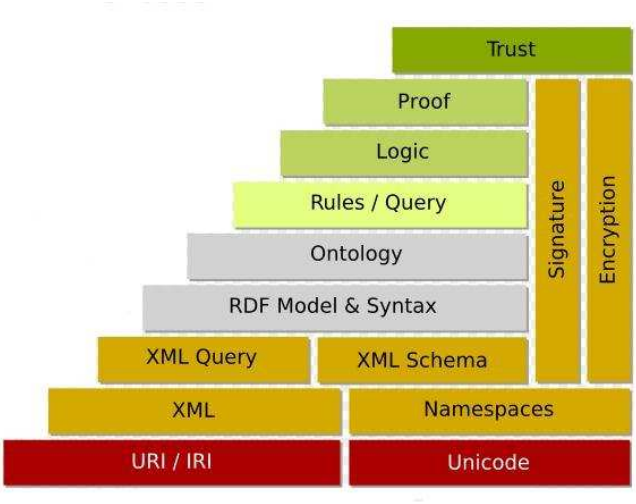


Figure 1: Semantic Web Technologies

3 REPOSITORY DEVELOPMENT

This section presents the development of the repository of knowledge for virtual communities. Six development stages are explained: Requirements Elicitation, Domain Modeling, Use Case Modeling, Architectural Design, Detailed Design, and Implementation. The repository is implemented using Semantic Web technologies, particularly the Resource Description Framework, the language for representing information about resources in the Web. The main advantage of the implementation approach is the representation of information in a standard way enabling interoperability with others repositories of knowledge through sharing and exchanging information.

3.1 Requirements Elicitation

The functional requirements for managing the repository are related with three main functions:

- *modifying the repository* - adding, removing and changing its components,
- *exploring the repository* - performing searches and browsing,
- *managing users* - administering user registrations and privileges.

Table 1 presents the requirements for modifying the repository. These requirements enable to create and remove type categories from the category hierarchy, register and remove resource types, and add and remove properties and statements. Type categories can be of two kinds: resource type categories - such as Conference, Book, Paper, etc; and data type categories - such as Integer, Text, Figure, etc. Users can add a type category by simply providing its name (CAT01), making the category a child of the *root* category. However, it is possible to create a category as a sub-category of an existing one (CAT02). In the latter case the user provides the name for the new category and the name of the parent category. Users can decide to remove categories (CAT03), although this operation can only be performed if no resources of the type category exist in the repository and no properties are defined with domain or range of that type category. In addition, properties can be defined by indicating its identifier name and the type categories for its domain and range - the domain must be a resource type category and the range may be both a resource or data type category. Two operations for defining

(PRO01) and removing (PRO02) properties are provided. A property can be removed only if there are no statements in the repository holding the property to be removed as predicate.

Table 1: Functional Requirements for Modifying the Repository

CAT01	Creating a type category
CAT02	Creating a type sub-category
CAT03	Removing an existing type category
PRO01	Defining a new property
PRO02	Removing an existing property
RES01	Registering a new resource to the repository
RES02	Removing an existing resource
STA01	Writing a new statement
STA02	Removing an existing statement

The repository is enriched by its users by incorporating new resources and discovering information and relationships about and among them. A new resource is registered by providing a name to identify it and the resource type category to which it belongs (RES01). It is possible to remove a resource (RES02) only in the case that no statements describing it or relating it to other resources exist in the repository. Knowledge acquired concerning a resource as well as relationships to other resources are added to the repository in the form of statements. A statement is added (STA01) providing the subject -resource about which the statement is reflecting information-, the predicate -property defined in the repository-, and the object -value of the property for the subject being described-. An operation for removing statements is also provided (STA02).

Table 2: Functional Requirements for Navigating the Repository

BRO01	Browsing categories
BRO02	Browsing resources
BRO03	Browsing properties
BRO04	Browsing statements
SEA01	Searching resources from a category
SEA02	Searching statements about a resource
SEA03	Searching statements with a predicate
SEA04	Searching properties with a particular domain
SEA05	Searching properties with a particular range

Requirements for exploring the repository are listed in Table 2 including operations for two main types of exploration: browsing the content and performing predefined searches. Browsing the repository content consists in obtaining a list of all the elements currently available in the repository. Users can browse the defined categories (BRO01), the set of registered resources (BRO02), the defined properties (BRO03), and the set of existing statements (BRO04). A group of predefined search operations is provided enabling to perform specific searches. Resources can be searched according to their category (SEA01); statements according to a particular resource they may have as subject (SEA02) or to a particular property as predicate (SEA03). Searches over properties can be executed according to their domain (SEA04) or their range (SEA05).

Table 3 presents the requirements for managing users. Users are responsible for updating the repository, therefore some restrictions and privileges are defined for them. Casual visitors of the

website including a link to the repository are allowed to explore the repository. However, operations involving changing the repository state is reserved for registered users, called members. Moreover, the creation and removal of categories can only be done by a special user playing the role of administrator. Managing such privileges implies the need to provide functions for registering (USE01) and un-registering (USE02) users; and for enabling users to log -in (USE03) and -out (USE04) of the application.

Table 3: Functional Requirements for Managing Users

USE01	Registering user
USE02	Un-registering user
USE03	Logging in to the application
USE04	Logging-out from the application

3.2 Domain Modeling

The development aim is to follow the RDF recommendation for implementing the repository. RDF, as a language, was designed for representing information about web resources. In RDF, resources are referenced using Uniform Resource Identifiers (URIs) [19], and described in terms of properties and values through statements. RDF statements are triples consisting of a *subject*, a *predicate* and an *object*. The *subject* is the resource being described by the statement; the *predicate* represents the property, characteristic or relation that is being assigned to the subject; and the *object* is the value of the mentioned property. Properties are described by assigning each its own URI and providing the domain and range of the property. The property domain identifies the resource type that will be described by such property, and the property range represents the data type identifying the allowed values for the property. While the domain must always be a resource type, the range can be a resource as well as a data type, thus providing greater flexibility.

The concepts explained above are modeled as follows: a repository is formed by the resources added by the users and the information gathered about these resources. The information about resources is stored by defining properties and statements. For modeling resource and data types, an abstract category type is introduced for representing both sorts of types.

The conceptual model is shown in Figure 2 depicting the main concepts of the repository and their relations. Two type categories exist: *Resource Type* and *Data Type Categories*. A *Type Category* is a subcategory of another category. *Resource Type Categories* provide the types for resources, while *Data Type Categories* provide types for simple data. *Resources* and *Data* are generalized in the class *Value*. A *Property* is composed of a *Range* and a *Domain*, both of them are categories. The *Domain* must be a *Resource Type Category* however the *Range* can be any *Type Category*. *Statements* have three components: *Subject*, *Predicate* and *Object*. The *Subject* is an instance of a *Resource*; the *Predicate* is a *Property* in the repository and the *Object* is a *Value* instance, i.e. a resource or plain data (depending on the definition of the predicate). A *Statement* describes a *Resource*, and one *Resource* can be described by many *Statements*. A *Repository* holds a set of *Resources*, a set of *Properties* and a set of *Statements* as well as a hierarchy of *Type Categories*. *Users* interact with the *Repository* and can be *Registered* or *Unregistered*. There is one special user among the registered users known as *Administrator*.

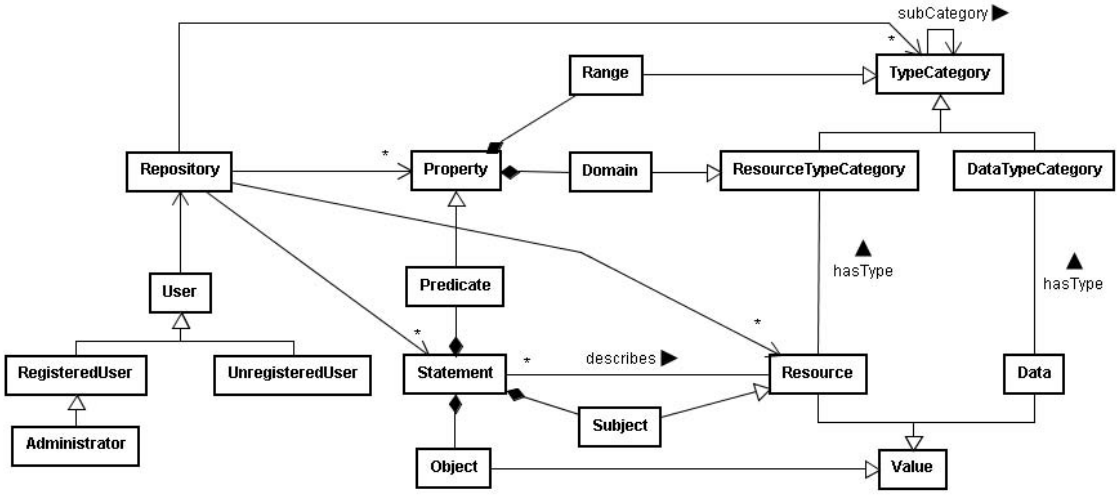
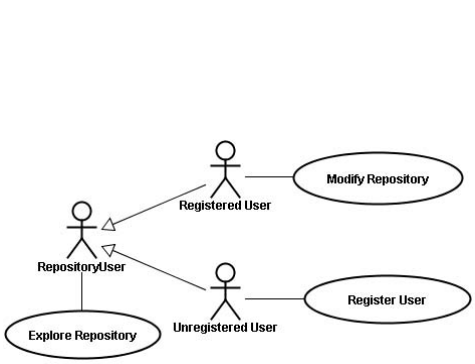


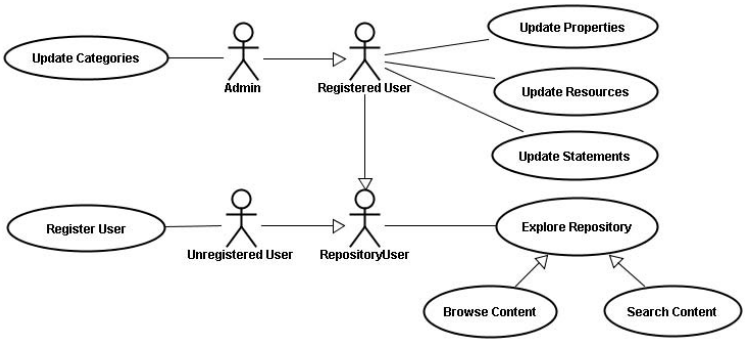
Figure 2: Conceptual Model

3.3 Use Case Modeling

This section presents the use case model which is depicted in Figure 3. Figure 3(a) presents the top level use case diagram showing the types of users interacting with the system and the available interactions. Every user (RepositoryUser) can use the repository to explore it (Explore Repository); not registered users (UnregisteredUser) can also make use of the registration feature (Register User); and every usage of the repository that implies changes over it (Modify Repository) can only be performed by registered users (Register User). Figure 3.3 shows Modify Repository and Explore Repository use cases in more details as well as the specialization of Registered User as the administrator (Admin) user.



(a) Top level use case



(b) Top level use case in more detail

Figure 3: Use Case Model

The Explore Repository use case involves two different use cases: Browse Content and Search Content. The Modify Repository use case has been broken down in four detailed functions: Update Categories, Update Properties, Update Resources and Update Statements. The first function can only be performed by the Admin user, while the others by any sort of Registered User.

3.4 Architectural Design

The application has a layered architecture composed of three tiers, as shown in Figure 4. The *Presentation Layer* includes the *JSP* pages used for displaying the repository content and supporting the interactions with the user. This layer requests services to the *Business Layer* who provides the corresponding functions and elements to populate the web pages. Thus, the *Business Layer* contains the components implementing the management of the repository and the management of users; providing functions for updating and searching the repository content and data about users. The *Persistency Layer* contains the component for managing the database. This component handles data persistency through a database, hence for storing and retrieving the repository content the *Business Layer* uses the *Persistency Layer*.

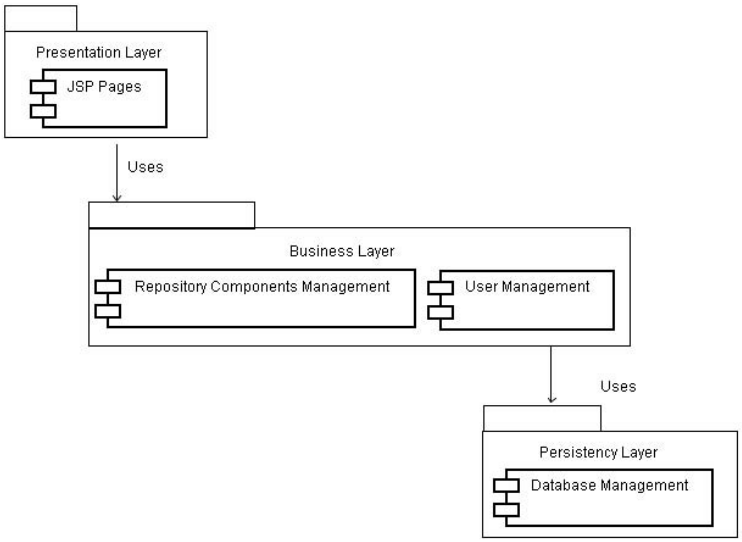


Figure 4: Architectural Static View

3.5 Detailed Design

The system design is shown in Figure 5. A static view of the system is provided by a design class diagram showing the classes implementing the main concepts described in the previous sections. A *Property* object can be identified by its name, and maintains the relationships with the the domain and range categories. A *Statement* keeps the triple relationship between the subject (*Resource*), predicate (*Property*) and object (*Value*). A *Resource* object only contains a name and the relationship to its own *Resource Type Category*. The *Repository* keeps one collection for each of these types of element: properties, statements and resources; plus two hierarchies of categories for resource type and for data type categories. The hierarchies are presented as a mapping from a category to the set of child categories. The root categories of both hierarchies are represented by the attributes resRoot and dataRoot.

3.6 Implementation

An initial implementation of this repository of knowledge is in place providing the repository services application for the virtual community of practice UNeGov.Net - Community of Practice for Electronic Governance [18]. The application is implemented in Java following the Apache Struts [16]

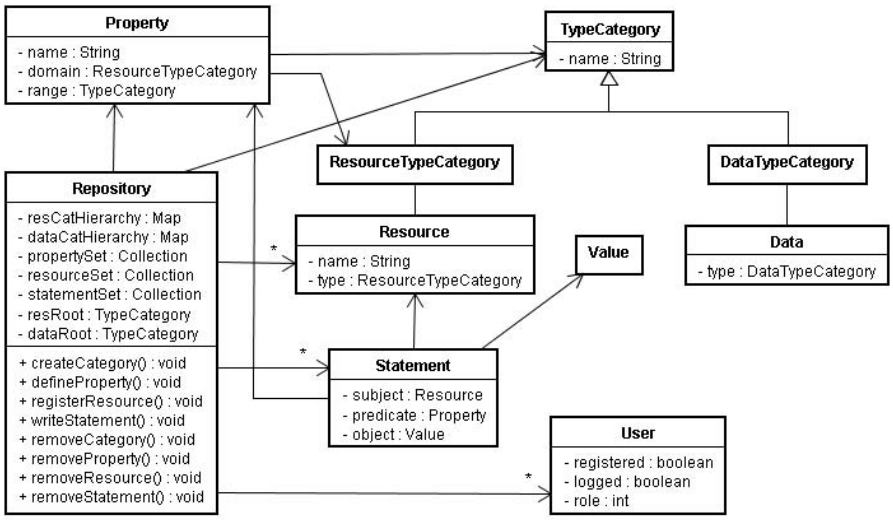


Figure 5: Design Class Diagram

framework. Persistence is obtained using MySQL [10] database, and Hibernate [6] is used for the object-relational mapping. The software is implemented as a Java Web Application and it needs to be deployed in a J2EE application server with a MySQL server. For the registration features to work as expected, a mail server is required for sending confirmation e-mails to the registered users.

4 APPLICATION

The presented repository of knowledge can be deployed as part of any portal or website. As mentioned before, a prototype has been deployed and is running as the repository of knowledge of UNeGov.Net - Community of Practice for Electronic Governance [18]. This version has been implemented including the following features:

a) User registration

b) User login and logout

c) Administration features

d) Browse content

e) Update content

Visitors arriving to the site can register themselves as members of the community, this process results in the user becoming a registered user of the community as well as becoming a resource of the repository under a special category created for members. Members is a sub-category of a more general category existing in the repository for general resource people. Once a person has registered as a member, he/she can log in into the system to update content, for instance defining new properties, registering new resources, modifying stored information about existing resources, writing statements, etc. Administrators of the site can also login into the site to perform operations reserved for them, like creating new categories, approving or denying the content updates requested by registered members, and assigning the administrator role to a registered user of the site. Browsing content allows any user

visiting the site to list the complete set of resources of the repository, or to filter the list by category. Properties can also be browsed, as well as the information regarding a particular resource, including the statements written about it. The left window in Figure 6 shows some of the currently available resources in the repository, while the right window shows how properties are displayed when browsed, together with the statements existing for a particular resource of category Paper.

Figure 6: Browsing the UneGov.Net Repository

5 CONCLUSIONS AND FUTURE WORK

The aim of this paper was to present a model and an implementation of a repository of knowledge for virtual communities based on Semantic Web technologies. This idea was motivated by the shortage of semantic web support for collaborative work on the web. The repository presented follows the Resource Description Framework standard and implements its main concepts. A detailed explanation of the design and development of the repository was presented and a concrete application of the system produced was shown: the Community of Practice for Electronic Governance (UNeGov.Net). The main advantage of this implementation is that it allows interoperability with other repositories or systems implementing Semantic Web technologies while keeping the design simple and clear for community users.

Semantic Web technologies are a subject of extensive interest among researchers, particularly in the area of knowledge management. Many applications and frameworks are being designed and built to use RDF and ontology languages like OWL, proving that the benefits from using these technologies are widely acknowledged. In [5] the author presents an ontology-based repository dedicated to software patterns, aiming to solve issues like establishing relationships between patterns and the dissemination of the patterns as well as the discovered relations among them. The work described in [15] pursues the introduction of ontologies for knowledge management in the field of web based learning to enable efficient reuse and sharing of knowledge. These approaches to knowledge management in the web, as well as our proposal, see ontologies as the backbone of the repositories being provided to the users to exploit and to rely on. They provide a solid tool to the automatization of knowledge discovery however such approaches do not provide the users the opportunity to be part of the creation of the vocabularies. The application presented in this work provides users with the mechanisms to enrich the repository through the addition and creation of new resources as well as through the creation of new ways to relate them and annotate them. It aims to reduce the shortage of support for these

kinds of procedures. As far as we know, there is no other application with such a general objective, promoting interoperability of resources through different communities.

Future work on this subject include considering the usage of ontologies to introduce the possibility of making inferences over the available information. In this sense, it will be necessary to represent, validate and extend our ontology using some OWL dialect.

Acknowledgments.

We wish to thank Tomasz Janowski for leadership, support and useful comments for developing this work. This work was partly supported by United Nations University - International Institute for Software Technology (UNU-IIST), Macao S.A.R. China, and Agencia Nacional de Promoción Científica, Argentina, through Project PAV-076/2003 "Intelligent Systems Supporting Productive Processes" sub-project: "Web Services and Intelligence for the WEB"

REFERENCES

[1] W3C. World Wide Web Consortium. http://www.w3.org/.

[2] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.

[3] Dan Brickley and Ramanathan V. Guha. RDF vocabulary description language 1.0: RDF schema. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

[4] Stefan Decker, Sergey Melnik, Frank van Harmelen, Dieter Fensel, Michel C. A. Klein, Jeen Broekstra, Michael Erdmann, and Ian Horrocks. The semantic web: The roles of XML and RDF. *IEEE Internet Computing*, 4(5):63–74, 2000.

[5] Scott Henninger. Using the semantic web to construct an ontology-based repository for software patterns. *Workshop on the State of the Art in Automated Software Engineering*, pages 18–22, June 2002.

[6] Hibernate. relational persistence. http://www.hibernate.org/.

[7] Graham Klyne and Jeremy J. Carroll. Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[8] Deborah L. McGuinness, Michael K. Smith, and Chris Welty. OWL web ontology language guide. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-owl-guide-20040210/.

[9] MusicBrainz. http://musicbrainz.org/.

[10] Mysql open source database. http://www.mysql.com/.

[11] Jean Paoli, C. M. Sperberg-McQueen, and Tim Bray. XML 1.0 recommendation. first edition of a recommendation, W3C, February 1998. http://www.w3.org/TR/1998/REC-xml-19980210.

[12] Constance Elise Porter. A typology of virtual communities: A multi-disciplinary foundation for future research. *Journal of Computer-Mediated Communication*, 10(1), 2004.

[13] RDF site summary (RSS) 1.0, December 2000. http://web.resource.org/rss/1.0/.

[14] SIOC project. semantically-interlinked online communities. http://sioc-project.org/.

[15] Bhavani Sridharan, Alexei Tretiakov, and Kinshuk. Application of ontology to knowledge management in web based learning. *Proceedings of IEEE International Conference on Advanced Learning Technologies*, pages 663–665, 2004.

[16] Struts. the apache software foundation. http://struts.apache.org/.

[17] The friend of a friend (FOAF) project. http://www.foaf-project.org/.

[18] UNeGov.Net. community of practice for electronic governance. http://www.unegov.net.

[19] Uniform resource identifier (URI): Generic syntax. http://gbiv.com/protocols/uri/rfc/rfc3986.html.

[20] Priscilla Walmsley and David C. Fallside. XML schema part 0: Primer second edition. W3C recommendation, W3C, October 2004. http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/.

Eligiendo Raíces para el Árbol de Aproximación Espacial

Alejandro Gómez, Verónica Ludueña y Nora Reyes
Departamento de Informática, Universidad Nacional de San Luis.
San Luis, 5700, Argentina
{*agomez,vlud,nreyes*}*@unsl.edu.ar*

Abstract

Many computational applications need to search information in a database. At the present time the predominance of multimedia databases does that the *similarity search* or *proximity search*, that is to look for elements of the database that are similar to a given query element, becomes a preponderant concept.

The Spatial Approximation Trees have shown to be competitive for similarity search in spaces with medium to high dimensionality ("difficult" spaces) or for queries with low selectivity. Nevertheless, for its construction its root was chosen randomly and it completely determines the tree, not only in its shape but also in its searching performance. Thus, our interest was to optimize searches in this data structure trying to choose the tree root in a way that the characteristics of indexed space can be reflected. We consider that, by this way, the data structure can adapt itself better to the dimension of the considered metric space, which results in more efficient similarity searches.

Keywords: similarity search, metric spaces, databases, algorithms.

Resumen

Muchas aplicaciones computacionales necesitan buscar información en una base de datos. En la actualidad el predominio de las bases de datos multimedia hace que la *búsqueda por similitud* o *búsqueda por proximidad*, es decir buscar elementos de la base de datos que sean similares a un elemento de consulta dado, se vuelva un concepto preponderante.

El Árbol de Aproximación Espacial ha demostrado ser muy competitivo para la búsqueda por similitud en espacios métricos de media a alta dimensionalidad (espacios "difíciles") o para responder a consultas con baja selectividad. Sin embargo, para su construcción se elegía su raíz al azar y ello determinaba completamente el árbol tanto en su forma como en su desempeño. Así, nuestro interés fue el de optimizar las búsquedas en dicha estructura tratando de que la raíz sea elegida de manera tal que refleje alguna de las características propias del espacio métrico a indexar. Creemos que de esta forma permitimos que la estructura se adapte mejor a la dimensión intrínseca del espacio métrico considerado, lo cual redunda en búsquedas más eficientes.

Keywords: búsqueda por similitud, espacios métricos, bases de datos, algoritmos.

1. INTRODUCCIÓN Y MOTIVACIÓN

Con la evolución de las tecnologías de información y comunicación, han surgido almacenamientos no estructurados de información. No sólo se consultan nuevos tipos de datos tales como texto libre, imágenes, audio y vídeo; sino que, en algunos casos, no se puede estructurar más la información en claves y registros. Estos tipos de datos son difíciles de estructurar para adecuarlos al concepto tradicional de búsqueda. Aún cuando sea posible una estructuración clásica, nuevas aplicaciones tales como la minería de datos requieren acceder a la base de datos por cualquier campo y no sólo los marcados como "claves". Así, han surgido aplicaciones en grandes bases de datos donde se desea buscar objetos similares. Este tipo de búsqueda se conoce con el nombre de *búsqueda aproximada* o *búsqueda por similitud* y tiene aplicaciones en numerosos campos.

Como en toda aplicación que realiza búsquedas, surge la necesidad de tener una respuesta rápida y adecuada. El planteo general del problema es: existe un universo $\mathbb{U}$ de *objetos* y una función de distancia positiva d: $\mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}^+$ definida entre ellos. Esta función de distancia satisface los axiomas que hacen que el par (U, d) sea un *espacio métrico*: positividad estricta ($d(x, y) = 0 \Leftrightarrow x = y$), simetría ($d(x, y) = d(y, x)$) y desigualdad triangular ($d(x, z) \leq d(x, y) + d(y, z)$). Mientras más "similares" sean dos objetos menor será la distancia entre ellos. Tenemos una *base de datos* finita $S \subseteq \mathbb{U}$ que puede ser preprocesada (v.g. para construir un índice). Luego, dado un nuevo objeto del universo (una *query* q), debemos recuperar todos los elementos similares que se encuentran en la base de datos. Existen dos consultas básicas de este tipo:

Búsqueda por rango: recuperar todos los elementos de S a distancia a lo más r de un elemento q dado.

Búsqueda de k vecinos más cercanos: dado q, recuperar los k elementos más cercanos a q en S.

La distancia se considera costosa de evaluar (por ejemplo, comparar dos huellas dactilares). Por lo tanto, es usual definir la complejidad de la búsqueda como el número de evaluaciones de distancia realizadas, dejando de lado otras componentes. Entonces, el objetivo es generar un índice que permita reducir al máxino el cálculo de distancias durante una búsqueda.

Un caso particular de este problema surge cuando el espacio es un conjunto D-dimensional de puntos y la función de distancia pertenece a la familia L_p de Minkowski: $L_p = (\sum_{1 \leq i \leq d} |x_i - y_i|^p)^{1/p}$. Existen métodos efectivos para buscar sobre espacios D-dimensionales, tales como *Kd–trees* [2, 3] o *R-trees* [8]; pero, para 20 dimensiones o más dejan de trabajar bien. Aunque nos dedicamos a espacios métricos generales, las soluciones planteadas son adecuadas para espacios D–dimensionales.

En [6, 4, 5] se muestra que el concepto de dimensionalidad intrínseca se puede entender aún en espacios métricos generales, se da una definición cuantitativa de ella y se muestra analíticamente la razón para la llamada "*maldición de la dimensionalidad*". Es interesante notar que el concepto de dimensionalidad está relacionado con la "*facilidad*" o "*dificultad*" para buscar en un espacio vectorial D-dimensional. Se dice que un espacio métrico general es más "difícil" (dimensión intrínseca más alta) que otro cuando su histograma de distancia es más concentrado. Esto hace que el trabajo de cualquier algoritmo de búsqueda por similitud sea más dificultoso. En el caso extremo tenemos un espacio donde $d(x, x) = 0$ y $\forall y \neq x, d(x, y) = 1$. En este caso, la consulta q debe ser exhaustivamente comparada contra cada elemento en el conjunto.

De las situaciones descriptas vemos que necesitamos contar con índices que permitan responder eficientemente a cada consulta. Existen numerosos métodos para preprocesar un conjunto a fin de reducir el número de evaluaciones de distancia y, en general, todos ellos se basan en dividir la base de datos, lo que se ha heredado de las ideas clásicas de *dividir para conquistar* y de la búsqueda de datos

típicos (v.g. árboles de búsqueda binaria). Uno de estos métodos que sigue un enfoque distinto y que ha demostrado ser eficiente es el *Árbol de Aproximación Espacial* (*SAT* por su sigla en inglés), ver [12], que se basa en la aproximación espacial. Esta estructura de datos es estática; es decir, necesita contar con los elementos de la base de datos de antemano para construir el índice que luego permita responder consultas por similitud. Nuestro trabajo consistió en optimizar esta estructura de datos a fin de mejorar, principalmente, su desempeño en las búsquedas.

El *SAT* ha demostrado ser muy competitivo en espacios métricos de media o alta dimensionalidad o para responder a consultas con baja selectividad. Sin embargo, para su construcción se elegía su raíz al azar y ello determinaba completamente el árbol tanto en su forma como en su desempeño [9, 13]. Así, nuestro interés fue el de optimizar las búsquedas en dicha estructura seleccionando de diferente manera la raíz del árbol, de forma tal que refleje alguna de las características propias del espacio métrico a indexar. Creemos que vale la pena tomarse el trabajo de elegir mejor la raíz y de esta forma permitir que la estructura se adapte mejor a la dimensión intrínseca del espacio métrico considerado, lo cual redunda en búsquedas más eficientes.

2. ÁRBOL DE APROXIMACIÓN ESPACIAL

El método de aproximación espacial comienza la búsqueda en algún punto del espacio y trata de acercarse "espacialmente" a la consulta q. La estructura de datos que utiliza este método de búsqueda es el *SAT*. Comenzaremos viendo en qué consiste la aproximación espacial.

Consideremos un espacio métrico $(\mathbb{U}, d)$ y a $S \subseteq \mathbb{U}$ que será nuestra base de datos. Para describir la idea básica de la aproximación espacial analizaremos las consultas de 1-*vecino más cercano* o 1-*NN*. Al buscar el vecino más cercano a q nos posicionamos en un elemento $a \in S$ elegido al azar, y nos acercarmos cada vez más y más a q moviéndonos a otro elemento $b \in S$ tal que $d(b, q) < d(a, q)$. Cuando no podamos acercarnos más a q será porque encontramos el elemento más cercano a él. Este proceso debe hacerse entre elementos que son "vecinos", o sea, estando en a sólo podemos acercarnos a q por medio de alguno de los vecinos de a, conjunto denotado de aquí en adelante como $N(a)$.

De acuerdo a esto, la estructura que más naturalmente se adapta a la restricción de movernos sólo hacia los vecinos de un elemento es un grafo dirigido, en donde los nodos se corresponden con los elementos de S y existen arcos entre un elemento y cada uno de sus vecinos. Los arcos denotan los posibles movimientos que podemos hacer para acercarnos a q. Concretamente, existe un arco desde a hasta b si es posible moverse de a a b en un único paso. En un espacio vectorial, el grafo minimal que buscamos corresponde a la triangulación clásica de Delaunay (un grafo donde los elementos que son vecinos de Voronoi están conectados). Así la respuesta ideal en términos de complejidad de espacio es el grafo de Delaunay (generalizado para espacios arbitrarios), y permitirá también búsquedas rápidas.

Lamentablemente no en todos los espacios métricos se puede construir un grafo de aproximación espacial. Esto nos lleva a realizar simplificaciones que nos permitirán resolver consultas del vecino más cercano a un $q \in S$ usando un árbol en lugar de un grafo, el *SAT*.

La construcción del *SAT* comienza con la selección, de manera *aleatoria*, de un elemento $a \in S$ que será la raíz o nodo de inicio del árbol, es decir el elemento desde el cual comienzan todas nuestras búsquedas. Luego se selecciona un conjunto adecuado de vecinos $N(a)$, que verifiquen:

Condición 2: (dados $a \in S$) $\forall x \in S, x \in N(a) \Leftrightarrow \forall y \in N(a) - \{x\},\ d(x, y) > d(x, a)$.

Así, los elementos de la base de datos que integran el conjunto $N(a)$ son aquellos que se encuentran más cerca de a que de cualquier otro elemento de la base de datos. Si comenzamos con el nodo raíz a debemos considerar la "bolsa" $B(a)$ para a, que inicialmente mantiene los elementos

en $S - \{a\}$. Se ordena el conjunto $B(a)$ por distancia a a de manera creciente, quedando primero el elemento que espacialmente es el más cercano a a. Sea $N(a)$ inicialmente vacío, se considera en orden cada elemento $b \in B(a)$; si ocurre que b es más cercano al nodo a que a cualquier elemento ya presente en $N(a)$, entonces se agrega b a $N(a)$. Analizados todos los $b \in B(a)$, se habrá determinado un conjunto $N(a)$ adecuado. Ahora hay que ocuparse de los elementos que fueron descartados como vecinos de a. Para ello se coloca cada elemento $b \in S - (a \cup N(a))$ en la bolsa $B(c)$ del elemento $c \in N(a)$ más cercano a b. Ya ubicado cada elemento en su bolsa, se considera a cada vecino de a como una raíz y se procesan recursivamente los elementos de su bolsa de la manera mencionada.

El algoritmo de la Figura 1 detalla este proceso de construcción. Para construir un *SAT* éste se debe invocar inicialmente con $BUILDTREE(a, S - a)$, en donde a será la raíz del árbol.

```
BuildTree(Nodo a, Conjunto de Elementos S)

1.   N(a) ← ∅          /* vecinos de a */
2.   R(a) ← 0          /* radio de cobertura */
3.   Ordenar S por distancia a a (más cercano primero)
4.   For v ∈ S
5.       R(a) ← máx(R(a), d(v,a))
6.       If ∀b ∈ N(a), d(v,a) < d(v,b) Then
7.           N(a) ← N(a) ∪ {v}
8.   For b ∈ N(a) S(b) ← ∅          /* subárboles */
9.   For v ∈ S − N(a)
10.      Sea c ∈ N(a) el que minimiza d(v,c)
11.      S(c) ← S(c) ∪ {v}
12.  For b ∈ N(a) BuildTree(b, S(b))          /* construir subarboles */
```

Figura 1: Algoritmo para construir un *SAT*.

Para evitar algunas comparaciones de distancia durante las búsquedas, se almacena en cada nodo b del árbol su radio de cobertura $R(b)$, que es la máxima distancia entre b y un elemento del subárbol del cual b es la raíz.

Una vez definida la estructura que nos va a permitir buscar por aproximación espacial, consideraremos las búsquedas por rango con radio r. La clave es que aunque $q \notin S$, la respuesta a la búsqueda son elementos $q' \in S$. La idea es usar el árbol para simular que buscamos un $q' \in S$ desconocido. Se sabe que $d(q, q') \leq r$ y que por la desigualdad triangular $\forall x \in U$, $d(x, q) \leq d(x, q') + d(q', q)$. Pero, como $d(q, q') \leq r$, podemos reemplazar $d(q, q')$ por r y manteniendo la desigualdad $d(x, q) - r \leq d(x, q')$. Por otro lado, por desigualdad triangular tenemos que $d(x, q') \leq d(x, q) + r$.

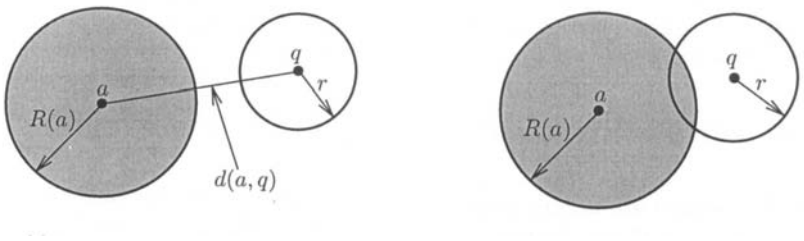
Si buscamos un q conocido, vamos directamente al vecino de a más cercano a q. Al ser q' desconocido no sabemos cuál es el vecino de a más cercano a q'. Entonces, estando en un nodo a, debemos determinar el vecino c más cercano a q entre los elementos de $\{a\} \cup N(a)$. Así, para $b \in \{a\} \cup N(a)$ se cumple que $d(c, q) < d(b, q)$. Pero es posible que $d(c, q') > d(b, q')$ y por lo tanto no encontraremos a q' si solamente buscamos dentro del subárbol de c. Entonces, debemos ingresar en todos los vecinos $b \in N(a)$ que cumplan que $d(q', b) \leq \text{mín } d(c, q) + 2r, c \in \{a\} \cup N(a)$. En otro caso, no estamos seguros de ello y debemos entrar al subárbol de b; porque el q' que estamos buscando puede diferir de q en a lo más r, por lo que podría haber sido insertado dentro de los subárboles de aquellos vecinos b.

Por lo tanto, lo que fue concebido como una búsqueda por aproximación espacial siguiendo un único camino, ahora se combina con backtracking para que busquemos por varios caminos.

Se puede mejorar el algoritmo si consideramos que cuando buscamos un elemento $q \in S$ seguimos un único camino desde la raíz hasta q pero sabemos que, si la búsqueda se encuentra en el nodo a del árbol, se puede evitar entrar en cualquier elemento $x \in N(a)$ tal que $d(q,x) > 2r + \min\{d(q,c), c \in \{a'\} \cup N(a'), a' \in A(a)\}$, donde $A(a)$ es el conjunto de ancestros de a (incluyendo a a) y $N(A(a)) = \bigcup_{a' \in A(a)} N(a')$, debido a que se puede mostrar, usando la desigualdad triangular, que ningún q' con $d(q,q') \leq r$ se pudo almacenar dentro de x.

Otra posibilidad de optimizar las búsquedas es usando la información almacenada sobre el radio de cobertura $R(a)$ de un nodo a, que nos permiten podar cierta parte del árbol, ya que no deberíamos ingresar a un subárbol con raíz a que verifica que $d(q,a) > R(a) + r$, porque esto implica que $d(q',a) > R(a)$ para cualquier q' tal que $d(q,q') \leq r$. Por la forma en que hemos definido $R(a)$, q' no puede encontrarse en el subárbol de a.

Las Figuras 2 y 3 ilustran las situaciones con las que nos podemos encontrar durante las búsquedas.



(a) **(b)**

Figura 2: En **(a)** se cumple que $d(a,q) > R(a) + r$, por lo que no debemos entrar en el subárbol de a ya que ninguno de sus elementos va a estar dentro del rango de la consulta. En **(b)** se cumple que $d(a,q) \leq R(a) + r$, así debemos entrar en el subárbol de a ya que es posible (no seguro) que alguno de sus elementos se encuentre dentro del rango.

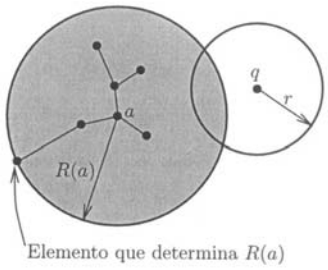


Figura 3: En este caso se cumple que $d(a,q) \leq R(a) + r$ y se observa que aunque debemos entrar en el subárbol de a, no existe ningún elemento en ese subárbol que esté dentro del rango de búsqueda.

En [12] se muestra que el *SAT* posee mejor desempeño en las búsquedas respecto de otras estructuras. El algoritmo de la Figura 4 describe el proceso de búsqueda que hemos explicado.

3. SELECCIÓN DE LA RAÍZ PARA EL SAT

Como se ya se mencionó, al construir el *SAT* éste queda completamente determinado por la elección de la raíz; es decir que, si para la misma base de datos tomáramos otro objeto como la raíz el árbol resultante sería distinto. Como esta elección se realiza al azar, es posible encontrar buenas raíces

```
BúsqRango (Nodo a, Query q, Radio r, Distancia d_mín)

1. If d(a,q) ≤ R(a) + r Then
2.     If d(a,q) ≤ r Then Informar a
3.     d_mín ← mín {d_min} ∪ {d(q,c), c ∈ N(a)}
4.     For b ∈ N(a)
5.         If d(b,q) ≤ d_mín + 2r Then BúsqRango (b, q, r, d_mín)
```

Figura 4: Algoritmo para buscar q con radio r en un *SAT*.

logrando que el árbol generado se comporte mejor principalmente durante las búsquedas, pero también podemos encontrar raíces poco convenientes. La Figura 5 muestra un ejemplo de un espacio y dos posibles árboles para dicho espacio generados a partir de la elección de diferentes raíces.

Sin embargo, elegir la raíz aleatoriamente es atractivo porque no hay costos adicionales, la elección es gratis dado que medimos nuestros costos en *cantidad de evaluaciones de la función de distancia*. Por otra parte, parece razonable utilizar información de la base de datos para seleccionar una buena raíz a costa de pagar un cierto costo en evaluaciones de distancia, esperando que éste se compense al disminuir principalmente los costos de las búsquedas, o al menos de la construcción.

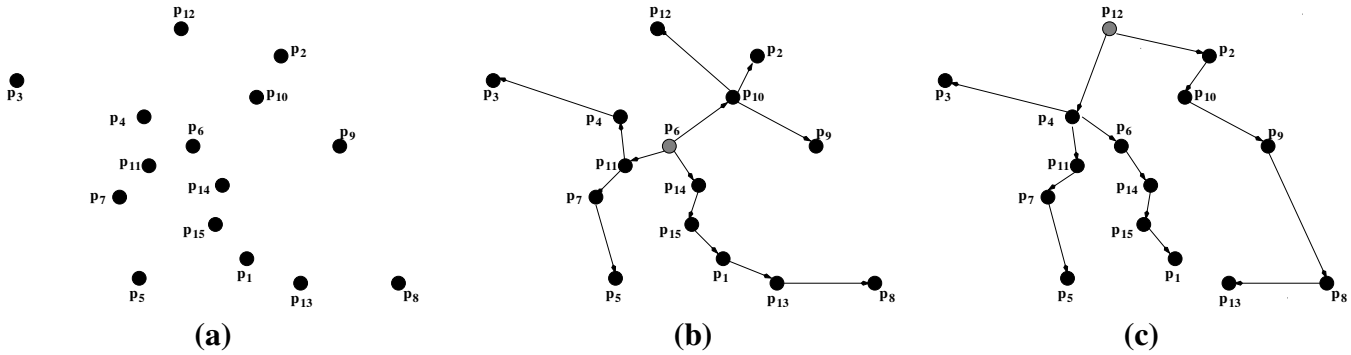


Figura 5: En (a) vemos un posible espacio de puntos, en (b) se muestra el *SAT* obtenido para ese espacio tomando como raíz al punto p_6, y en (c) se ve el *SAT* generado eligiendo a p_{12} como raíz.

3.1. Métodos de Selección de la Raíz

Aquí asumiremos que un elemento es mejor raíz que otro si en el árbol, que desde él se genere, se obtienen mejores costos principalmente en las búsquedas. El énfasis estaría puesto en las búsquedas, porque al ser el *SAT* una estructura estática, no admite ni inserciones ni eliminaciones, la selección de la raíz se realiza una única vez antes de construir el árbol.

Hemos estudiado diferentes métodos para seleccionar un elemento como raíz en el *SAT* y a través del análisis experimental determinamos qué método obtiene los mejores costos de búsqueda. Aunque nos interesarán más los costos de las búsquedas, destacamos para cada método el costo de la elección de la raíz medido en cantidad de evaluaciones de la función de distancia. En algunos de ellos subyacen ideas de teoría de grafos, otros son adaptaciones de métodos usados en otras estructuras para efectuar procesos similares y el método *CSA*, propuesto en [13], pensado específicamente para el *SAT*.

Los métodos considerados son: Método CSA (Centroid Selection Algorithm), Aleatorio, Sampling, M-LB-DIST (Maximum Lower Bound on DISTance), mM-LB-RAD (minimun Maximun Lower Bound RADius), mM-AVG-RAD (minimun Maximun Average RADius).

Los últimos cuatro métodos se han adaptado desde métodos utilizados en el *M–tree* para seleccionar el centro de un nuevo nodo luego de una operación de *split* [7]. Fueron adaptados de manera tal que pudieran usarse en el *SAT* para elegir una raíz, manteniendo los costos razonables. Para identificar fácilmente cada método del *M–tree* hemos mantenido sus nombres originales.

A continuación se hará una breve descripción de cada uno. Para el análisis de los costos de selección de la raíz, y suponiendo que $S \subseteq \mathbb{U}$ es nuestra base de datos, consideramos $N = |S|$.

3.1.1. Método CSA (Centroid Selection Algorithm)

Este método propuesto para elegir la raíz del *SAT*, fue presentado por Penarrieta, Morriberón y Cuadros-Vargas en [13]. La idea subyacente es que un elemento que podría ser una buena raíz para el *SAT* sería aquél que es cercano al centroide ideal de la base de datos. Está basado en el algoritmo *HF* presentado en la Familia Omni [14] y se refleja en el siguiente algoritmo:

1. Seleccionar aleatoriamente un elemento $s \in S$.
2. Encontrar el elemento más lejano e_1 de s.
3. A partir de e_1 determinar su elemento más distante e_2.
4. Seleccionar como raíz al elemento c que satisfaga que $d(e_1, c) \simeq (e_2, c)$ y que minimice $|d(e_1, e_2) - (d(e_1, c) + d(e_2, c))|$.

Este último paso es muy importante porque pueden existir varios candidatos para c, pero se elige aquél que minimice el perímetro del triángulo formado por e_1, e_2 y c y que además no se encuentre retirado del centro del triángulo. La Figura 6 permite visualizar los pasos a desarrollar en el algoritmo. Este método necesita realizar $3N$ evaluaciones de distancia.

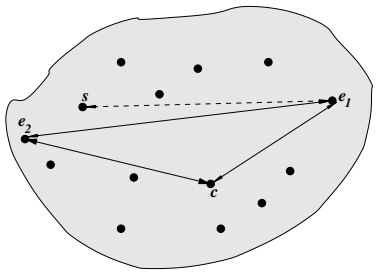


Figura 6: Idea geométrica que da origen al Método CSA.

3.1.2. Método Sampling

Es una política aleatoria que trabaja sobre una muestra de objetos de S de tamaño fijo $s > 1$. Seleccionamos al azar s objetos de la base de datos, y elegimos como raíz al elemento que tenga *menor distancia máxima* a los $s - 1$ objetos restantes. Es fácil observar que en este caso se realizan $\frac{s * (s-1)}{2}$ cálculos de la función de distancia, lo que nos daría un costo de $O(s^2)$. Con el fin de mantener este costo lineal respecto de la cardinalidad de S, elegimos a $s = \sqrt{N}$; así el método debe realizar N cálculos de distancia para elegir la raíz.

3.1.3. Método M-LB-DIST (Maximum Lower Bound on DISTance)

Esta política difiere de la anterior en que necesita utilizar todo el conjunto de elementos. Elegimos un objeto al azar y calculamos las distancias a todos los demás objetos, luego seleccionamos el objeto más distante como raíz. Formalmente, elegimos un elemento $x \in S$ aleatoriamente y determinamos el $y \in S$ tal que $d(x, y) = \max_{z \in S}\{d(x, z)\}$. Claramente este proceso realiza N evaluaciones de distancia para obtener la raíz.

3.1.4. Método mM-LB-RAD (Minimun Maximun Lower Bound Radius)

Se elige un objeto $x \in S$ al azar, y se calculan las distancias de todos los elementos $z \in S$ a x, luego elegimos como raíz el objeto y para el que $\text{máx}_{z \in S}\{|d(y,x) - d(z,x)|\}$ sea mínimo. Este criterio selecciona un elemento y que está a una distancia intermedia de x. Aunque la manera en que se elige a un elemento y como raíz es más compleja que en el método anterior, la cantidad de cálculos de distancia es también N.

3.1.5. Método mM-AVG-RAD (Minimun Maximun Average Bound Radius)

Este método surgió como una alternativa al anterior y trata de ser neutral con respecto a la información de los límites superior e inferior, tomando el promedio y aplicando el criterio de selección $\text{mín} - \text{máx}$. Así, elegimos nuevamente al azar un elemento $x \in S$, calculamos para todo $z \in S$ la distancia $d(x,z)$ y elegimos un elemento y como raíz tal que $\text{máx}_{z \in S}\{(d(y,x) + d(z,x))/2\}$ sea mínimo. Este método también necesita N cálculos de distancia para poder seleccionar la raíz.

4. RESULTADOS EXPERIMENTALES

Para analizar el desempeño de cada uno de los métodos de selección de la raíz del *SAT* se realizaron experimentos sobre diferentes espacios métricos, para así poder estudiar el comportamiento de cada uno de ellos de manera más objetiva. Consideramos principalmente los costos de las búsquedas, sin embargo analizamos también qué sucede con los costos de construcción y por ende con los costos de selección de la raíz. Los espacios métricos considerados son los siguientes:

Espacio de Palabras: formado por un diccionario de 69.069 palabras del Inglés y la *distancia de edición*, que es el mínimo número de inserciones, supresiones y/o reemplazos de caracteres necesarios para hacer una palabra igual a otra. Esta distancia es útil para tratar con deletreo o reconocimiento de palabras, recuperación de texto, escritura y errores en reconocimiento óptico de caracteres (OCR).

Espacio de Vectores de Imágenes de la NASA: un conjunto de 40.700 vectores de características de 20 componentes, generados desde imágenes descargadas desde el sitio de la NASA [1]. La *distancia Euclídea* es utilizada como la función de distancia sobre este espacio.

Espacio de Histogramas de Imágenes: un conjunto de 112.682 vectores, formado por histogramas de color 8-D de 112 componentes para cada imagen [2]. La distancia debía ser de la familia de Minkowski, así se eligió la *distancia Euclídea* como la alternativa más simple y significativa.

Espacio de Documentos: formado por un conjunto de 1.265 documentos y la función de *distancia coseno*, muy utilizada en recuperación de la información [1]. Los documentos son obtenidos de la colección TREC- 3 [3] y son vistos como vectores (cada término de los documentos es una coordenada).

Espacio de Vectores de Coordenadas de Dimensión 15: es un espacio integrado por 100.000 vectores de dimensión 15, con componentes reales en el cubo unitario y *distancia Euclídea*. Para este espacio hemos generado 100.000 puntos al azar con distribución uniforme en el espacio $[0,1]^{15}$. Lo trataremos como a un espacio métrico, desechando la información que brindan las coordenadas. Esto nos permite controlar la dimensionalidad exacta con la que trabajaremos, lo que no es fácil en un espacio métrico general o si los vectores provienen desde una situación real.

Los experimentos consisten en ejecutar la construcción del *SAT* para los diversos algoritmos de selección de raíces y en cada espacio métrico considerado, y luego buscar en ellos. Para la construcción

[1] Disponible en: `http://www.dimacs.rutgers.edu/Challenges/Sixth/software.html`

[2] Disponible en `http://www.dbs.informatik.uni-muenchen.de/~seidl/DATA/histo112.112682.gz`

[3] Disponible en: `http://trec.nist.gov`

del índice se usa el $90\,\%$ de los elementos del espacio métrico. El $10\,\%$ restante se utilizan como objetos de búsqueda. Realizamos varias ejecuciones de cada experimento sobre permutaciones distintas de la base de datos; por lo tanto, los resultados exhibidos corresponden a valores medios obtenidos.

Para el Espacio de Palabras en Inglés, con función de distancia discreta, los radios de búsqueda utilizados en los experimentos fueron de 1 a 4, que recuperan en promedio aproximadamente el 0.00003 %, el 0.00037 %, el 0.00326 % y el 0.01757 % respectivamente. Los radios de búsqueda utilizados en los espacios con distancia continua, son los siguientes:

a) Espacio de Vectores de Imágenes de la NASA: 0.605740, 0.78 y 1.009.

b) Espacio de Histogramas de Imágenes: 0.051768, 0.082514 y 0.131163.

c) Espacio de Documentos: 0.189441, 0.222466 y 0.600048.

d) Espacio de Vectores de Dimensión 15: 0.686576, 0.833130 y 1.019767.

Estos radios recuperan en promedio aproximadamente el 0.01 %, 0.1 %, 1 % elementos de la base de datos, respectivamente.

4.1. Comparación General de los Métodos

Comparamos todas las técnicas en cada espacio particular, para ver cuáles son las que mostraron menor número de evaluaciones de distancia en las consultas. Se agregan en las gráficas resultados de métodos de Mínimas y Máximas distancias que no fueron considerados como métodos alternativos en este trabajo por ser demasiado costosos, pero que sirven como referencia para los demás métodos.

4.1.1. Espacio de Palabras

De la comparación se observa que aquí, de los métodos considerados, los que tienen menor costo para las búsquedas son CSA y Sampling. Cabe destacar que el método de selección de la raíz Aleatorio no parece obtener un buen árbol, desde el punto de vista de las búsquedas, dado que casi todas las demás maneras de elegir la raíz, logran superarlo para los cuatro radios considerados. La Figura 7 muestra estos resultados.

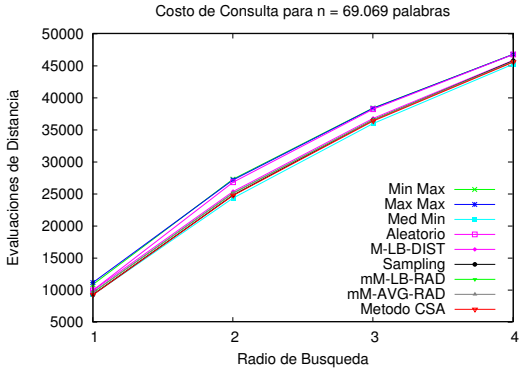


Figura 7: Comparación de los costos de búsqueda para todos los métodos sobre el Espacio de Palabras.

4.1.2. Espacio Vectores de Imágenes de la NASA

En este espacio el método que tuvo mejor desempeño en las búsquedas es el M-LB-DIST. Los métodos mM-LB-RAD y mM-AVG-RAD obtuvieron idénticos desempeños. Nuevamente aquí se ve que el método Aleatorio no produce el mejor árbol para las búsquedas. En la Figura 8 podemos apreciar este analisis.

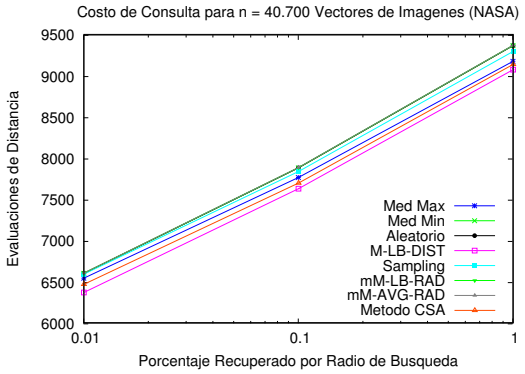


Figura 8: Comparación de los costos de búsqueda para todos los métodos sobre el Espacio de Vectores de Imágenes de la NASA.

4.1.3. Espacio de Histogramas de Imágenes

En este espacio métrico se comprobó que el método más económico en las búsquedas fue el Aleatorio y a continuación, con muy poca diferencia, la técnica CSA. Es llamativo que sea el método Aleatorio el que produce el menor costo en cantidad de evaluaciones en las búsquedas, siendo éste un algoritmo de costo $O(1)$ en cantidad de cálculos de distancia, y sumamente simple en su elaboración. Sin duda las características de este espacio hacen que resulte infructuoso usar nuestras técnicas, pero claramente no ha sido el caso en la mayoría de los espacios utilizados. La Figura 9 refleja estos hechos.

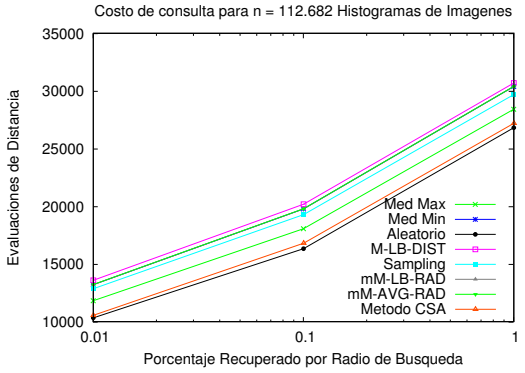


Figura 9: Comparación de los costos de búsqueda para todos los métodos sobre el Espacio de Histogramas de Imágenes.

4.1.4. Espacio de Documentos

En este espacio, el mejor método en las búsquedas para todos los radios y que además obtuvo un buen costo en la construcción es el método CSA, aunque las diferencias entre los distintos métodos no son demasiado significativas, ya que entre el mejor y el peor costo de búsqueda para cada radio la diferencia es cercana a 50 evaluaciones de distancia. En la Figura 10 es posible observar esta situación.

4.1.5. Espacio Vectores de Dimensión 15

En los experimentos se observa que existe una gran similitud entre los costos de las búsquedas de casi todos los métodos, no existiendo prácticamente diferenciación entre ellos. El único método que se destaca como menos costoso, por escasa diferencia de evaluaciones de los demás métodos, es CSA en todos los radios considerados. Sin embargo, aunque CSA haya mostrado mejor desempeño en las búsquedas, resultó ser el peor en cuanto a costos de construcción. Si no se está dispuesto a pagar tanto

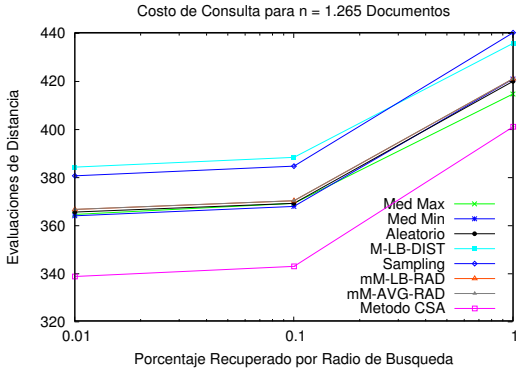


Figura 10: Comparación de los costos de búsqueda para todos los métodos sobre el Espacio de Documentos.

en la construcción de la estructura una muy buena alternativa sería el método de Sampling porque su desempeño en las búsquedas es muy similar y su costo de construcción es mucho menor. La Figura 11 permite observar gráficamente estos resultados.

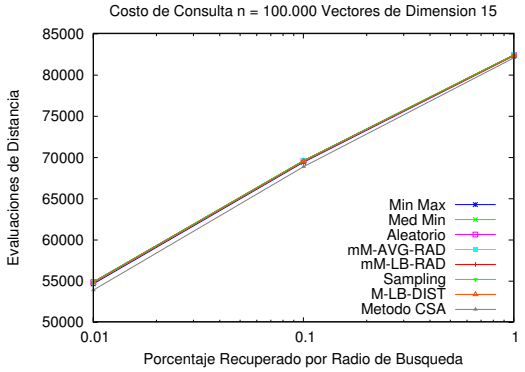


Figura 11: Comparación de los costos de búsqueda para todos los métodos sobre el Espacio de Vectores de Dimensión 15.

5. CONCLUSIONES

El *Árbol de Aproximación Espacial* (*SAT*) ha mostrado ser muy competitivo en espacios métricos de media a alta dimensionalidad o respondiendo a consultas con baja selectividad [12][9]. Un análisis detallado muestra que la selección de la raíz del árbol es uno de los procesos que se podrían optimizar, dado que éste es el que determina completamente al *SAT* y originalmente se realizaba al azar. Otros autores se han dedicado a este tema [13]. Sin embargo creemos que nuestro trabajo brinda más información debido a que se estudiaron y compararon más métodos, la mayoría de los cuales fueron diseñados o adaptados por nosotros para esta estructura, y se realizaron más experimentos.

Hemos conseguido construir un árbol más eficiente para las búsquedas, sólo pagando algunos cálculos de distancia adicionales respecto del *SAT* original y manteniendo la correctitud de la estructura. Demostramos experimentalmente que un mayor conocimiento sobre el espacio métrico particular permite mejorar la estructura, y que los algoritmos de selección de la raíz propuestos logran *mejorar* tanto costos de búsqueda como los costos de construcción del *SAT* original. Aunque nuestros resultados son aplicables principalmente al *SAT*, algunos podrían serlo en otras estructuras arbóreas.

De la misma manera que se mejoró la versión estática de *SAT* eligiendo la raíz de modo diferente al original, se podría intentar adaptar alguna de las técnicas a la versión dinámica del *SAT* [10, 11].

Hasta ahora el *SAT* trabaja en memoria principal; la posibilidad que la selección de una mejor raíz resulte en un árbol más balanceado, podría hacer que su almacenamiento en memoria secundaria fuera más eficiente.

REFERENCIAS

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[2] J. Bentley. Multidimensional binary search trees used for associative searching. *Comm. of the ACM*, 18(9):509–517, 1975.

[3] J. Bentley. Multidimensional binary search trees in database applications. *IEEE Trans. on Software Engineering*, 5(4):333–340, 1979.

[4] E. Chávez and G. Navarro. Measuring the dimensionality of general metric spaces. Technical Report TR/DCC-00-1, Dept. of Computer Science, University of Chile, 2000.

[5] E. Chávez and G. Navarro. Towards measuring the searching complexity of metric spaces. In *Proc. Mexican Computing Meeting*, volume II, pages 969–978, Aguascalientes, México, 2001. Sociedad Mexicana de Ciencias de la Computación.

[6] E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, September 2001.

[7] P. Ciaccia, M. Patella, and P. Zezula. M-tree: an efficient access method for similarity search in metric spaces. In *Proc. of the 23rd Conference on Very Large Databases (VLDB'97)*, pages 426–435, 1997.

[8] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 47–57, 1984.

[9] G. Hjaltason and H. Samet. Improved search heuristics for the sa-tree. *Pattern Recognition Letters*, 24(15):2785–2795, 2003.

[10] G. Navarro and N. Reyes. Fully dynamic spatial approximation trees. In *Proceedings of the 9th International Symposium on String Processing and Information Retrieval (SPIRE 2002)*, LNCS 2476, pages 254–270. Springer, 2002.

[11] G. Navarro and N. Reyes. Dynamic spatial approximation trees. *ACM Journal of Experimental Algorithmics (JEA)*, 2007. To appear.

[12] Gonzalo Navarro. Searching in metric spaces by spatial approximation. *The Very Large Databases Journal (VLDBJ)*, 11(1):28–46, 2002.

[13] José Peñarrieta, Patricio Morriberón, and Ernesto Cuadros-Vargas. Distributed spatial approximation tree- sat$*$. In M. Marin and G. Acuña, editors, *Actas de la XXXII Conferencia Latinoamericana de Informatica (CD ROM)(CLEI2006)*, 2006.

[14] R.F. Santos Filho, A. Traina, C. Traina Jr., and C. Faloutsos. Similarity search without tears: The omni family of all-purpose access methods. In *Proc. of the 17th International Conference on Data Engineering*, pages 623–630, 2001.

Estrategias de Reparación para Sitios Web Incompletos*

M. Alpuente[1], D. Ballis[2], M. Falaschi[3], P. Ojeda[1], and D. Romero[1]

[1] DSIC, Universidad Politécnica de Valencia
Camino de Vera s/n, Apdo. 22012, 46071 Valencia, Spain.
{alpuente, pojeda, dromero}@dsic.upv.es

[2] Dip. Matematica e Informatica
Via delle Scienze 206, 33100 Udine, Italy.
demis@dimi.uniud.it

[3] Dip. di Scienze Matematiche e Informatiche
Pian dei Mantellini 44, 53100 Siena, Italy.
moreno.falaschi@unisi.it

Abstract. El desarrollo y mantenimiento de sitios Web no es un tarea fácil. Garantizar que la información de un sitio Web es consistente resulta cada vez más dificil, por ello los administradores Web necesitan de mecanismos que ayuden a reparar estas inconsistencias. En este trabajo, completamos la metodología de reparación semiautomática de sitios Web erróneos presentada en un trabajo anterior con el tratamiento de los errores de completitud. Comenzamos con la categorización de dichos errores mediante dos nuevos órdenes que definimos sobre el conjunto de errores de completitud obtenidos en el proceso de verificación. Utilizando estos órdenes, definimos sendas estrategias capaces de minimizar la cantidad de acciones de reparación a realizarse para generar un sitio Web completo con respecto a su especificación.

Keywords: Verificación y reparación de sitios Web, reescritura, simulación, reescritura parcial, estrategias de reparación.

1 Introducción

Los sistemas software para la web se han convertido en un instrumento insustituible de la moderna sociedad de la información: hacen posible el intercambio de información de forma rápida y a escala global; constituyen el medio natural de las grandes transacciones financieras; permiten acceder de forma rápida y selectiva a grandes volúmenes de información especializada, etc. Todo esto ha incrementado la complejidad de los sitios web y puesto de manifiesto la necesidad de asistir a los administradores de sistemas web en la detección y reparación de las posibles incorrecciones o inconsistencias.

En este escenario, es esencial el desarrollo de métodos, modelos y herramientas que se apliquen a la verificación formal de sistemas y servicios web, y que permitan no sólo detectar automáticamente posibles errores en los enlaces o en la estructura sino también en la semántica de los mismos. Los fallos de calidad deben ser detectados con precisión para que, de esta manera, se puedan aplicar estrategias de reparación (idealmente) automáticas que permitan obtener sitios web que sean correctos y completos con respecto a una especificación o modelo conceptual de los mismos.

* This work has been partially supported by the EU (FEDER) and Spanish MEC TIN-2004-7943-C04-02 project, the Generalitat Valenciana under grant GV06/285, and Integrated Action Hispano-Alemana HA2006-0007. Daniel Romero is also supported by ALFA grant LERNet AML/19.0902/97/0666/II-0472-FA. Pedro Ojeda is also supported by the Generalitat Valenciana under FPI grant BFPI/2007/076.

En el marco de la reparación existen también diferentes líneas de investigación incipientes. En [17], se presenta un marco para reparar inconsistencias en documentos distribuidos, como un complemento a la herramienta de verificación xlinkit [11]. La principal contribución es la semántica que pone en correspondencia el lenguaje lógico de primer orden de xlinkit a un catálogo de acciones reparadoras que pueden ser usadas para corregir de forma interactiva las violaciones de reglas, aunque este trabajo no predice si la ejecución de la acción puede provocar una nueva violación de dichas reglas. Tampoco es posible detectar si dos expresiones que formulan un requerimiento para el sitio Web son incompatibles. Similarmente, en [18, 20] se presenta una extensión para CDET [19]. Esta extensión incluye un mecanismo para eliminar inconsistencias de un conjunto de documentos interrelacionados. Primero se genera un grafo dirigido acíclico (*DAG*) que representa las relaciones entre documentos, y entonces se derivan las reparaciones directamente de este grafo. En este caso, las reglas temporales soportan interferencias y la compatibilidad de las reparaciones sí es tenida en cuenta. Desafortunadamente, esta compatibilidad resulta muy costosa de comprobar en reglas temporales. Ambas aproximaciones se basan en técnicas del campo de las bases de datos activas [8]. Otras investigaciones recientes en este campo se centran en la derivación de reglas activas que, automáticamente, lancen acciones reparadoras que conducen a un estado consistente después de cada actualización [16].

En trabajos previos [6, 2], presentamos un marco para la verificación automática de sitios Web. Dicho marco permite especificar restricciones de integridad para un sitio Web y entonces comprobar automáticamente si estas restricciones se satisfacen. El marco proporciona un lenguaje de especificación que permite definir propiedades sintácticas así como semánticas de un sitio Web. Como resultado de la verificación se identifican dos tipos de errores: error de corrección (*correctness error*) y error de completitud (*completeness error*).

Tomando como base el marco de verificación de [2], en [3] hemos formulado también una metodología para la corrección del código erróneo. En ella, partiendo de una categorización de los diferentes tipos de error que pueden encontrarse en un sitio Web respecto a su especificación inicial, clasificamos las distintas clases de acciones de reparación que pueden ejecutarse para corregir cada error dado.

Cómo complemento a este trabajo, en [7] realizamos un análisis sistemático sobre la relación entre los errores de corrección detectados por nuestra metodología y, de esta manera, definimos dos estrategias, una para minimizar el número de acciones de reparación a ser ejecutadas y la otra para minimizar la información a ser cambiada o eliminada sobre el sitio Web. El resultado de la fase de corrección es un sitio Web correcto w.r.t. a su especificación aunque no completo.

Nuestra contribución. En este trabajo, completamos nuestra metodología de reparación [3, 6] para considerar y eliminar los errores de completitud. Empezamos analizando la relación entre los errores de completitud e introducimos dos órdenes ($\preceq_{inf}$ y $\preceq^{sup}$) que se corresponden con dos criterios diferentes de priorización u ordenación de los errores. Aprovechando este análisis formalizamos dos operaciones de reparación: *repairByDelete* la cual se apoya en el orden $\preceq_{inf}$ y *repairByInsert* la cual se apoya en el orden $\preceq^{sup}$. Con la utilización de estas operaciones, definimos dos estrategias que reducen la cantidad de acciones necesarias para reparar un sitio Web. Esto nos permite mejorar nuestro sistema de reparación, al tiempo que generar un sitio que cumple totalmente sus especificaciones.

Estructura del trabajo. El resto del trabajo está estructurado como sigue. La sección 2 resume algunas definiciones y notaciones preliminares. En la sección 3, recordamos un método simple de transformación de documentos XHTML/XML en términos de Herbrand, mostramos la noción de simulación utilizada para reconocer patrones dentro de un sitio Web y presentamos

el lenguaje de especificación de sitios Web. En la sección 4 formalizamos los errores de completitud que pueden ser detectados tras el proceso de verificación y presentamos dos operaciones que permiten su reparación. La sección 5 proporciona un análisis de la dependencia entre los errores de completitud, mientras en la sección 6 desarrollamos las estrategias de reparación de un sitio Web incompleto al tiempo que demostramos que la reparación no introduce nuevos errores. La sección 7 presenta las conclusiones del trabajo.

2 Preliminares

Decimos que un conjunto finito de símbolos es un *alfabeto*. Dado el alfabeto A, A^* denota el conjunto de todas las secuencias finitas de elementos sobre A. La igualdad sintáctica entre objetos se representa como $\equiv$.

Denotamos por $\mathcal{V}$ un conjunto infinito de variables y Σ denota un conjunto de símbolos de función, o *signatura*. Consideramos las signaturas de aridad variable como en [12] (i.e., signaturas en las cuales los símbolos tienen aridad no especificada, es decir, pueden tener cualquier número de argumentos a continuación). $\tau(\Sigma, \mathcal{V})$ y $\tau(\Sigma)$ denotan el *álgebra de términos con variables* y el *álgebra de términos básicos* construidas en $\Sigma \cup \mathcal{V}$ y Σ, respectivamente. Los términos se consideran árboles etiquetados de la manera usual.

Las posiciones son representadas por secuencias de números naturales que denotan el camino de acceso a un término. La secuencia vacía Λ denota la posición raíz. Con la notación $w_1.w_2$, se denota la concatenación de la posición w_1 y la posición w_2. Las posiciones son ordenadas por el orden de prefijos, que es, dada las posiciones w_1, w_2, $w_1 \leq w_2$ si existe una posición x s.t. $w_1.x = w_2$.

Dado $S \subseteq \Sigma \cup \mathcal{V}$, $O_S(t)$ denota el conjunto de posiciones de un término t que tenga como raíz al símbolo S. Por otra parte, para una posición x, $\{x\}.O_S(t) = \{x.w \mid w \in O_S(t)\}$.

Con $t|_v$ representamos el subtérmino cuya raíz es v de t. $t[r]_v$ es el término t con el subtérmino cuya raíz es v sustituido por el término r.

Una *sustitución* $\sigma \equiv \{X_1/t_1, X_2/t_2, \ldots\}$ es una aplicación del conjunto de variables $\mathcal{V}$ en el conjunto de términos $\tau(\Sigma, \mathcal{V})$ que satisface las siguientes condiciones: $(i) X_i \neq X_j$, si $i \neq j$, $(ii) X_i\sigma = t_i, i = 1, ..n$, y $(iii) X\sigma = X$, para $X \in \mathcal{V} \setminus \{X_1, ..., X_n\}$. Con ε representamos una *sustitución vacía*. Dada una sustitución σ, el *dominio* de σ es el conjunto $Dom(\sigma) = \{X | X\sigma \not\equiv X\}$. Dadas las sustituciones σ_1 y σ_2, tal que $Dom(\sigma_1) \subseteq Dom(\sigma_2)$, por σ_1/σ_2, definimos la sustitución $\{X/t \in \sigma_1 | X \in Dom(\sigma_1) \setminus Dom(\sigma_2)\} \cup \{X/t \in \sigma_2 | X \in Dom(\sigma_1) \cap Dom(\sigma_2)\} \cup \{X/X | X \notin Dom(\sigma_1)\}$. Una *instancia* de un término t se define como $t\sigma$, donde σ es una sustitución. Con $Var(s)$ representamos el conjunto de variables que aparecen en el objeto sintáctico s.

Los sistemas de rescritura de términos proporcionan un modelo computacional adecuado para los lenguajes funcionales. En consecuencia, seguimos un marco estándar de rescritura de términos para formalizar nuestra propuesta (ver [5, 14]).

Un *sistema de rescritura de términos* (TRS para abreviar) es un par (Σ, R), donde Σ es una signatura y R es un conjunto finito de reglas de reducciones (o rescrituras) de la forma $\lambda \to \rho, \lambda, \rho \in \tau(\Sigma, \mathcal{V}), \lambda \notin \mathcal{V}$ and $Var(\rho) \subseteq Var(\lambda)$. A menudo escribimos R en vez de (Σ, R). Un paso de rescritura es la aplicación de una regla de rescritura a una expresión. Un término s se rescribe a un término t via $r \in R$, $s \to_r t$ (or $s \to_R t$), si existe una posición $u \in O_\Sigma(s)$, $r \equiv \lambda \to \rho$, y una sustitución σ tal que $s_{|u} \equiv \lambda\sigma$ y $t \equiv s[\rho\sigma]_u$. Cuando no haya riesgo de confusión, omitiremos cualquier subíndice (por ejemplo $s \to t$). Un término s está en una *forma irreducible* (o forma normal) w.r.t. R si no existe un término t s.t. $s \to_R t$. Un t es la

forma irreducible de s w.r.t. R (en símbolos $s \rightarrow_R^! t$) si $s \rightarrow_R^* t$ y t es irreducible. Decimos que un TRS R es *terminante*, si no existe una secuencia infinita $t_1 \rightarrow_R t_2 \rightarrow_R ....$ Un TRS R es *confluente* si, para todos los términos s, t_1, t_2, tal que $s \rightarrow_R^* t_1$ y $s \rightarrow_R^* t_2$, existe un término t s.t. $t_1 \rightarrow_R^* t$ y $t_2 \rightarrow_R^* t$. Cuando R es terminante y confluente, se denomina *canónico*. En un TRS canónico, cada término de entrada t puede ser reducido a una única forma *irreducible*.

3 Denotación de sitios Web

Sean dos alfabetos T y $\mathcal{T}ag$. Representaremos el conjunto T^* por $\mathcal{T}ext$. Llamaremos a un objeto $t \in \mathcal{T}ag$ elemento *tag*, y a un objeto $w \in \mathcal{T}ext$ elemento *text*. Las páginas Web se proporcionan con una estructura de árbol; de esta manera, pueden ser traducidas de forma directa a términos ordinarios de un álgebra de términos $\tau(\mathcal{T}ext \cup \mathcal{T}ag)$ dada [2]. Note que los atributos tag XML/XHTML pueden considerarse elementos etiquetados comunes y, por tanto, traducidos de la misma forma que éstos. De esta manera, un *sitio Web* puede ser representado como un conjunto finito de términos *ground* $\{p_1 ... p_n\}$.

En lo siguiente, también consideraremos los términos del álgebra de términos no-ground $\tau(\mathcal{T}ext \cup \mathcal{T}ag, \mathcal{V})$, los cuales pueden contener variables. Un elemento $s \in \tau(\mathcal{T}ext \cup \mathcal{T}ag, \mathcal{V})$ es llamado "plantilla de página Web" (*Web page template*).

En nuestra metodología usamos las plantillas de páginas Web para especificar patrones de páginas Web incompletas o ausentes en el sitio Web. Para mecanizar la detección de estos patrones dentro de un sitio Web, empleamos un mecanismo de simulación de árboles, que pasamos a describir en la siguiente sección. En las Figuras 1(a) y 1(b), se ve un ejemplo de transformación de una página Web a un elemento del álgebra de términos.

```
<book>                              book(
  <title>El Alquimista</title>       title(El Alquimista),
  <author>Coelho</author>            author(Coelho),          book(code(X),author(Y))
  <year>2002</year>                  year(2002),
  <code>PC02</code>                  code(PC02)
</book>                             )
          (a)                               (b)                        (c)
```

Fig. 1. Ejemplo de transformación de una página Web a un elemento del álgebra de términos, y una plantilla de página Web.

3.1 Simulación

La noción de *simulación* nos permite analizar y extraer la estructura parcial de una página Web sujeta a verificación. Las simulaciones se han utilizado en varios trabajos relacionados con la consulta, transformación y verificación de datos semiestructurados (ej. [1, 10, 13]). Para simplificar nuestra exposición, no consideramos la semántica de las etiquetas; esto requeriría introducir ontologías, lo cual queda fuera del alcance de este trabajo.

La noción de simulación, $\trianglelefteq$, es una adaptación de la relación de *embedding* de Kruskal [9], ignonardo la regla de *diving*[4] [15].

[4] La regla de *diving* permite "saltar" parte del término del lado derecho de la relación $\trianglelefteq$. Formalmente, $s \trianglelefteq f(t_1, \ldots t_n)$, si $s \trianglelefteq t_i$, para algún i.

Definition 1 (simulación).
La relación de simulación $\trianglelefteq \subseteq \tau(\mathcal{T}ext \cup \mathcal{T}ag) \times \tau(\mathcal{T}ext \cup \mathcal{T}ag)$ *sobre páginas Web se define como sigue:*

$$\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_m) \trianglelefteq \mathtt{g}(\mathtt{s}_1, \ldots, \mathtt{s}_n) \textit{ sii } \mathtt{f} \equiv \mathtt{g} \textit{ y}$$
$$\mathtt{t}_i \trianglelefteq \mathtt{s}_{\pi(i)}, \textit{ para } \mathtt{i} = 1, \ldots, \mathtt{m}, \textit{ y}$$
$$\textit{alguna función inyectiva}$$
$$\pi : \{1, \ldots, \mathtt{m}\} \rightarrow \{1, \ldots, \mathtt{n}\}.$$

Dadas dos páginas Web s_1 y s_2, si $s_1 \trianglelefteq s_2$ se dice que s_1 *simula* (o *es embebida* o *reconocida* dentro de) s_2. También se dice que s_2 *embebe* s_1. Note que, en la Definición 1, para el caso donde $\mathtt{m}$ es 0 se tiene $\mathtt{c} \trianglelefteq \mathtt{c}$ para cada símbolo constante $\mathtt{c}$. Note también que $s_1 \not\trianglelefteq s_2$ si s_1 o s_2 contiene variables.

Un ejemplo de simulación se observa en la Figura 1: la plantilla de página Web de la Figura 1(c) puede ser reconocida dentro de la estructura del (fragmento de) página Web de la Figura 1(b).

3.2 Lenguaje de especificación

En esta sección, describiremos brevemente el lenguaje de especificación Web dado en [2]. Una especificación Web es una tripla (I_N, I_M, R), donde I_N e I_M son conjuntos finitos de reglas de corrección (*correctness rule*) y reglas de completitud (*completeness rule*), y R un conjunto de funciones auxiliares.

El conjunto I_N describe restricciones para detectar páginas Web erroneas (*correctNess rules*). Un análisis de optimización sobre los errores de corrección detectados se presentó en [7], y está fuera del alcance de este trabajo.

El tercer conjunto de reglas I_M especifica propiedades para detectar páginas Web incompletas/ausentes (*coMpleteness rules*). Una regla de completitud está definida como $l \rightharpoonup r \ \langle \mathtt{q} \rangle$, donde l y r son términos y $\mathtt{q} \in \{\mathtt{E}, \mathtt{A}\}$. Las reglas de completitud de una especifiación Web formalizan el requerimiento de que alguna información deba estar incluida en todas o algunas de las páginas del sitio Web. Los atributos $\langle \mathtt{A} \rangle$ y $\langle \mathtt{E} \rangle$ se usan para distinguir, respectivamente, reglas universales (*universal rule*) de las existenciales (*existential rule*), como explicamos mas adelante. La parte derecha de las reglas de completitud r puede contener funciones, las cuales están definidas en R y, algunos símbolos, pueden estar marcados por medio del símbolo $\sharp$. La información marcada en r se usa para seleccionar el subconjunto del sitio Web sobre el cual se verificará la condición formalizada por r.

Intuitivamente, la interpretación de una regla de completitud universal (resp. una regla de completitud existencial) w.r.t. un sitio Web W es la siguiente: si (una instancia de) l es "reconocida" en W, (una instancia de) la forma irreducible de r debe también ser "reconocida" en todas (resp. algunas de) las páginas Web que embeban (una instancia de) la estructura marcada de r. Una operación auxiliar *mark* es utilizada para obtener el subconjunto de páginas Web de W que se ajustan a la estructura de un término marcado r, por ejemplo, $mark(\sharp f(h(X)), \{p_1, p_2\}) = \{p_1\}$, donde $p_1 = f(m(a))$ y $p_2 = h(g(b))$.

3.3 Reescritura parcial

La reescritura parcial [2] nos permite extraer de una página Web s, usando una regla $l \rightharpoonup r$, una parte (subtérmino) de s que es simulada por alguna instancia *ground* de l, replazando

s por una instancia *ground* de r. Mas formalmente, sea s, $\mathtt{t} \in \tau(\mathcal{T}ext \cup \mathcal{T}ag, \mathcal{V})$. Entonces, s *reescribe parcialmente* a $\mathtt{t}$ por medio de la regla $l \rightharpoonup r$ y la sustitución σ sii existe una posisición $u \in O_{\mathcal{T}ag}(s)$ s.t. (i) $l\sigma \trianglelefteq s_{|u}$, y (ii) $\mathtt{t} = \mathit{Reduce}(r\sigma,R)$, donde la función $\mathit{Reduce}(x, R)$ computa por reescritura de términos estándar la forma irreducible de x en R. Note que el contexto de la expresión reducible seleccionada $s_{|u}$ es omitido después de cada paso de reescritura. Por $\mathtt{s} \rightharpoonup_I \mathtt{t}$, denotamos que $\mathtt{s}$ se reescribe parcialmente a $\mathtt{t}$ usando alguna regla que pertenece al conjunto I.

4 Errores y acciones de reparación

El lenguaje de especificación Web, junto con la técnica de verificación y validación de propiedades formales sobre sitios Web, permite detectar páginas Web incompletas o ausentes en un sitio Web, obteniendo así un conjunto de errores que denominaremos "errores de completitud" (*completeness error*). Estos errores representan la información inconsistente o ausente en el sitio Web.

Podemos distinguir tres tipos de errores de completitud: *Missing page* (o página ausente), cuando una expresión no aparece en el sitio Web; *Universal completenes error* (o error universal), *Existential completeness error* (o error existencial). Un error *universal* (resp. *existential*) corresponde a la satisfacción de una condición *universal* (resp. *existential*) del lenguaje de especificación.

Estos tres tipos de errores (M, A, E) pueden ser detectados por reescritura parcial sobre las páginas Web y la especificación I_M.

Más formalmente tenemos la siguiente definición.

Definition 2 (completeness error). *Sea W un sitio Web, (I_N, I_M, R) una especificación Web, y $c \in \{M, A, E\}$. Sea $q \in \{A, E\}$. Un error de completitud en W es la tupla $e \equiv (s_0 \rightharpoonup^*_{I_M} s_{n-1} \rightharpoonup_{I_M} s_n, P, c)$ que satisface:*

i) *Existe una sustitución σ s.t. $r\sigma = s_n$ para alguna $l \rightharpoonup r\langle \mathtt{q} \rangle \in I_M$.*
ii) *Existe una sustitución σ' s.t. $l'\sigma' = s_0$ para alguna $l' \rightharpoonup r'\langle \mathtt{q}' \rangle \in I_M$.*
iii) *Existe $p \in W$ s.t. $s_0 \trianglelefteq p$.*
iv) *$P = \{p \mid p \in \mathit{mark}(r, W)$ y $s_n \not\trianglelefteq p\}$.*
v) *$c = (M$ si $P = \emptyset)$ o $(\mathtt{q}$ si $P \neq \emptyset)$.*

En la definición 2, $s_0 \rightharpoonup^*_{I_M} \rightharpoonup s_{n-1} \rightharpoonup_{I_M} s_n$ representa la secuencia de pasos de reescritura hasta detectar el término s_n que no satisface la regla (por cuestión de simplicidad escribiremos $s_0 \rightharpoonup \ldots \rightharpoonup s_n$). Denotamos por P el conjunto de páginas Web incompletas, mientras que c es la clase o tipo del error detectado (M *missing page*, A *universal error*, y E *existential error*). Note que, para un error de tipo *Missing Page* el conjunto P es vacío.

Denotamos con $\mathbb{E}_M(W)$ (o simplemente $\mathbb{E}_M$), al conjunto de todos los errores de completitud detectados en un sitio Web W. Cuando $|\mathbb{E}_M(W)| = 0$ decimos que W está libre de errores de completitud o simplemente W está reparado.

Con el objetivo de reparar un sitio Web que contiene errores, en [3] definimos tres acciones reparadoras básicas de corrección que permiten eliminar la inconsistencia y/o añadir la información ausente. Las acciones consideradas son las siguientes:

- **delete**(p, t) elimina todas las ocurrencias del término t en de la página Web p, y devuelve la página Web modificada.

– **insert**(p, w, t) modifica la página Web p añadiendo el término t en $p_{|w}$.
– **add**(p, W) añade la página Web p al sitio Web W.

Note que las acciones reparadoras *add* e *insert* introducen nueva información en el sitio Web que puede contener/provocar nuevos errores. Por ello, es importante restringir la información que se puede añadir. Del mismo modo, la ejecución de la acción *delete* puede inducir nuevos errores de completitud. Para prevenir estas situaciones, en [3] formalizamos las propiedades *safe* y *acceptable* que garantizan el comportamiento seguro de estas acciones.

A groso modo, estas propiedades nos aseguran, respectivamente, que no se introducen nuevos errores, y que la cantidad de errores en el sitio Web decrece tras la ejecución de cada acción reparadora. Estas propiedades garantizan la terminación del proceso de corrección.

A lo largo de este trabajo, asuminos que la aplicación de las acciones reparadoras es siempre *acceptable*. Un sitio Web está reparado si $\mathbb{E}_M = \phi$.

4.1 Reparando errores de completitud

En esta sección se formulan las operaciones para reparar un sitio Web que contenga errores de completitud. Sin pérdida de generalidad, se asume que el sitio Web W es incompleto pero correcto w.r.t. una especificación Web (I_N, I_M, R). Esto se justifica por la posibilidad de asumir el haber ejecutado previamente la metodología de reparación de la corrección de [3]. Tal suposición permite diseñar una metodología reparadora la cual "completa" el sitio Web sin introducir información incorrecta. Como dijimos en la sección anterior, cualquier error $e \in \mathbb{E}_M(W)$ puede ser reparado añadiendo la información ausente o eliminando el dato que lo produjo. De igual forma, consideramos *acceptable* a cualquier acción que se ejecute, lo cual se puede comprobar fácilmente ejecutando la metodología de verificación de [2].

A continuación describimos cómo es posible reparar los errores de completitud detectados en un sitio Web. En primer lugar, veremos cómo añadir la información necesaria; a continuación, describiremos cómo eliminar la información incompleta que provocó el error.

4.2 Añadiendo información

Dependiendo del tipo de error de completitud, tenemos dos posibles acciones reparadoras a ser ejecutadas, **add**(p, W) e **insert**(p, w, t). La acción *add* se usa cuando tenemos un *missing page error* mientras la acción *insert* se ejecuta cuando el error de completitud es *universal error* o *existential error*. A continuación, analizaremos como reparar cada uno de ellos.

Missing Page error. Sea $e \equiv (s_0 \rightharpoonup \ldots \rightharpoonup s_n, P, q)$ un *missing page error*, donde $P = W$ y $q = M$. Para reparar e añadiemos al sitio Web W una página Web p que embeba la expresión $\mathtt{s_n}$. De esta manera tenemos

$$W = W \cup \{\textbf{add}(p, W)\}, \text{ donde } \mathtt{s_n} \trianglelefteq p_{|w} \text{ para algún } w \in O_{\mathcal{T}ag}(p).$$

Existential error.. Sea $e \equiv (s_0 \rightharpoonup \ldots \rightharpoonup s_n, P, q)$ un *existential error*, donde $q \equiv E$. Reparamos el error insertando el término s_n en una página arbitraria p tal que $p \in P$. De esta manera tenemos

$$W = W \setminus \{p\} \cup \{\textbf{insert}(p, w, s_n)\}, \text{ donde } \mathtt{s_n} \trianglelefteq p \text{ para algún } w \in O_{\mathcal{T}ag}(p).$$

Universal error Sea $e \equiv (s_0 \rightharpoonup \ldots \rightharpoonup s_n, P, q)$ un *universal error*, donde $q \equiv A$. Reparamos el error insertando el término s_n en cada página Web $p \in P$. De esta manera, el sitio Web W se transforma como sigue

$$W = W \setminus \{p\} \cup \{\mathbf{insert}(p, w, s_n)\} \quad \forall p \in P, \text{ y } \exists w \in O_{\mathcal{T}ag}(p)$$

Note que la información añadida para reparar un error podría casualmente ser la misma que sirva para reparar otro error. Postponemos a la Sección 5 el análisis de la posible relación entre los errores.

4.3 Eliminando información incompleta

En algunas situaciones es más conveniente eliminar la información incompleta; en particular, esta opción puede ser muy usada cuando se tiene información desactualizada. La principal idea es eliminar aquella información del sitio Web que causa el error de completitud sin necesidad de remplazar la nueva por información que satisfaga las especificaciones. El tratamiento de la información incompleta es independiente del tipo de error (M, A, E) que estemos analizando; por lo tanto, la información ausente se computa de la misma manera para los tres tipos de errores.

A continuación, damos la definición de la operación *repairByDelete*.

Definition 3 (*repairByDelete*)**.** *Dado un sitio Web W y un error de completitud $e \equiv (s_0 \rightharpoonup \ldots \rightharpoonup s_n, P, q)$, el sitio Web W se transforma aplicando acciones de borrado de la siguiente manera.*

$$\begin{aligned} repairByDelete(e, W) = &\{p \in W \mid s_0 \not\trianglelefteq p_{|w}, \forall\, w \in O_{\mathcal{T}ag}(p)\} \cup \\ &\{\mathbf{delete}(p, s_0) \mid p \in W, s_0 \trianglelefteq p_{|w}, w \in O_{\mathcal{T}ag}(p)\} \end{aligned}$$

En otras palabras, en la Definición 3 se elimina de todas las páginas Web la ocurrencia del término s_0 que inicia la secuencia de reescritura. Note que, si en la secuencia de reescritura de dos errores tenemos el mismo término inicial, es posible que la reparación de un error corrija de manera automática a otro.

5 Dependencia entre errores de completitud

Un sitio Web puede contener varios errores de completitud que podrían estar de alguna manera conectados. Por otro lado, con la ejecución de una operación reparadora es posible reparar más de un error. A continuación, analizamos cómo es la dependencia entre errores de completitud.

En primer lugar definimos dos órdenes parciales sobre el conjunto de errores $\mathbb{E}_M$, que están inducidos por los términos de la secuencia de reescritura parcial que llevan a la manifestación del error.

Definition 4. *(orden inducido por los inferiores) Sean $e_1 \equiv (s_0 \rightharpoonup \ldots \rightharpoonup s_n, P_1, q_1)$ y $e_2 \equiv (t_0 \rightharpoonup \ldots \rightharpoonup t_m, P_2, q_2)$ dos errores de completitud en $\mathbb{E}_M(W)$. Entonces,*

$$e_1 \preceq_{inf} e_2 \textit{ sii } s_0 \trianglelefteq t_0.$$

Diremos que un error $e \in \mathbb{E}_M(W)$ es minimal *w.r.t. $\preceq_{inf}$, sii no existe e' s.t. $e' \preceq_{inf} e$ y $e' \not\equiv e$.*

Definition 5. *(orden inducido por los superiores) Sean* $e_1 \equiv (s_0 \rightharpoonup \ldots \rightharpoonup s_n, P_1, q_1)$ *y* $e_2 \equiv (t_0 \rightharpoonup \ldots \rightharpoonup t_m, P_2, q_2)$ *dos errores de completitud en* $\mathbb{E}_M(W)$*. Entonces,*

$$e_1 \preceq^{sup} e_2 \text{ sii } s_n \trianglelefteq t_m.$$

En la Definición 4 se comparan los errores observando la relación de simulación que existe entre los términos que iniciaron la secuencia de reescritura de cada error. En la Definición 5, se observan en cambio los términos finales de esta secuencia. Decimos que e_1 y e_2 no son comparables w.r.t. $\preceq_{inf}$ (resp. $\preceq^{sup}$) sii $e_1 \not\preceq_{inf} e_2$ (resp. $e_1 \not\preceq^{sup} e_2$) y $e_2 \not\preceq_{inf} e_1$ (resp. $e_2 \not\preceq^{sup} e_1$).

Explotando estas dos definiciones sobre los errores obtenemos la siguiente proposición, que establece que reparar el menor de los errores (e_1) w.r.t. la relación $\preceq_{inf}$, utilizando la operación *repairByDelete*, permite reparar de manera automática el resto de los errores que están relacionados con e_1 según el órden $\preceq_{inf}$.

Proposition 1. *Sea* W *un sitio Web, y* $e_i \in \mathbb{E}_M(W), i = 1 \ldots m$*, y sea* $e_1 \preceq_{inf} \ldots \preceq_{inf} e_m$*. Entonces, realizando la acción reparadora* $repairByDelete(e_1, W)$ *se reparan todos los errores* $e_1, \ldots, e_m$ *en* W*.*

Proof. Si se cumple la relación $e_1 \preceq_{inf} \ldots \preceq_{inf} e_m$, por la Definición 4 tenemos $s_{1_0} \trianglelefteq s_{2_0} \trianglelefteq \ldots \trianglelefteq s_{m_0}$. Por tanto, al ejecutar $repairByDelete(e_1, W)$ se elimina el término s_{1_0} de W (por la relación $\trianglelefteq$, también se elimina un subtérmino de s_{2_0} y así sucesivamente hasta s_{m_0}). Junto con ellos se eliminan también los respectivos errores. Note que esta proposición es independiente del tipo que sean los errores (*Missing page, Universal, o Existential error*).

Veamos ahora cómo reparar el sitio Web añadiendo la información ausente y siendo $e_1 \preceq^{sup} e_2$. ¿Es posible en esta situación reparar de manera automática más de un error?. A continuación, profundizaremos el análisis sobre la relación entre errores de completitud con el objetivo de responder a esta pregunta.

En primer lugar, veremos algunas consideraciones necesarias:

- El orden de errores de completitud $\preceq^{sup}$ permite conocer cuál es el error que, al ser reparado, añadirá más información al sitio Web.
- El orden para tratar los errores debe ir de mayor a menor w.r.t. $\preceq^{sup}$; de esta manera, cuando se repare un error se añade información útil para los errores menores.
- Si e_n (es el mayor error w.r.t. $\preceq^{sup}$) es un error de *missing page* y/o existencial, de manera automática se reparan todos los errores de *missing page* y/o existenciales inferiores. Esto es debido a que, de una sola vez, se añade información que embebe a la información ausente indicada por los demás errores. Note que reparar un error universal actuará de la misma manera sobre los errores *missing page* y/o existenciales inferiores.
- Cuando un error universal es reparado, es posible que páginas que pertenezcan a otro error universal inferiror sean reparadas, y por lo dicho anteriormente, no pertenecen más al error.
- Una vez aplicada alguna acción reparadora en una página, no es necesario volver aplicarle otra acción en un error inferior en el orden $\preceq^{sup}$.

Estas consideraciones, están expresadas en el Algoritmo 1, que describe el procedimiento $repairByInsert$ que permite reparar errores de completitud ordenados por la relación $\preceq^{sup}$.

Los resultados obtenidos en la Proposición 1 y el Algoritmo 1, permiten una optimización obvia del marco de reparación, que formalizamos en las estrategias de reparación presentadas a continuación.

Algorithm 1 Procedimiento para reparar errores de completitud ordenados por la relación $\preceq^{sup}$.

Require:
$\mathbb{E} = e_i \in \mathbb{E}_M(W), i = 1, \dots m$, and $[e_1 \preceq^{sup} \dots \preceq^{sup} e_m]$
Un sitio Web W
Ensure:
$W \mid \forall e \in \mathbb{E}, e \notin \mathbb{E}_M(W)$

```
1: procedure repairByInsert (𝔼, W)
2:   P_R = {} // conjunto de páginas reparadas
3:   for i ← m to 1 do
4:     (s_0 ⇀ ... ⇀ s_n, P, q) ← e_i
5:     if q = M and P_R = {} then
6:       W ← W ∪ {add(s_n, W)}
7:       P_R ← P_R ∪ {s_n}
8:     else if q = E and P_R = {} then
9:       p ← element(P) // obtener una página
10:      p' ← insert(p, w, s_n) // w es una posición arbitraria en p
11:      W ← W \ {p} ∪ {p'}
12:      P_R ← P_R ∪ {p}
13:    else if q = A then
14:      P_Aux ← P \ P_R
15:      for all p ∈ P_Aux do
16:        p' ← insert(p, w, s_n) // w es una posición arbitraria en p
17:        W ← W \ {p} ∪ {p'}
18:        P_R ← P_R ∪ {p}
19:      end for
20:    end if
21:  end for
22: end procedure
```

6 Estrategias de reparación

Cómo explicamos en la Sección 4, es posible reparar un error de completitud por la ejecución de alguna acción reparadora. Por (e, a) denotamoa el par que contiene una acción reparadora a que corrije el error e. Además, por la notación $W' = a(e, W)$ especificamos la ejecución de la acción reparadora a sobre el sitio Web W, la cual retorna el sitio Web W' con el error e reparado.

Llamaremos *estrategia de reparación* a la ejecución de una secuencia de acciones reparadoras que permitan reparar todos los errores detectados en un sitio Web. Mas formalmente tenemos.

Definition 6 (estrategia de reparación). *Sea W un sitio Web y sea $\mathbb{E}_M(W) = \{e_1, \dots, e_n\}$ el conjunto de errores de completitud en W. Una* estrategia de reparación *para W es la secuencia $[(e_1, a_1), \dots, (e_n, a_n)]$, donde $a_1, \dots, a_n$ son acciones reparadoras s.t.*

$$(i)\ W_0 = W;$$
$$(ii)\ W_i = a_i(e_i, W_{i-1})\ \forall i, 1 \leq i \leq n;$$

y entonces, $\mathbb{E}_M(W_n) = \emptyset$.

En la Sección 5 vimos cómo, al reparar un error de completitud, es posible reparar de manera automática otro error. Este hecho nos sugiere que no es necesario ejecutar una acción reparadora por cada error detectado en un sitio Web. A continuación, presentamos dos posibles estrategias de reparación. En la primera, el objetivo es reducir la cantidad de información a eliminar para obtener un sitio Web libre de errores; en la segunda, en cambio, se persigue reducir la cantidad de información que se debe añadir. En ambos casos, sólo es necesario reparar un subconjunto de los errores del sitio Web.

6.1 Estrategia *reduce-delete-actions*

La relación $\preceq_{inf}$ define un orden parcial sobre el conjunto de errores de completitud $\mathbb{E}_M$. Por otro lado, en la Proposición 1, vimos que reparar un error *minimal* w.r.t. $\preceq_{inf}$ por medio de la operación *repairByDelete* permite reparar los demás errores relacionados con él según este orden. Es claro ver que si tenemos dos errores *minimales* e_1 y e_2 no comparables w.r.t. $\preceq_{inf}$ es necesario reparar ambos errores.

Definition 7 (estrategia *reduce-delete-actions*). *Sea W un sitio Web y sea $\mathbb{E}_M(W)$ el conjunto de errores de completitud de W. Una* estrategia de reparación *(o* $\mathbb{RDA}$-strategy*) que permite reducir las acciones de eliminación para W es*

$$[(e_1, repairByDelete), \ldots, (e_n, repairByDelete)],$$
$$\textit{donde } \forall i, 1 \leq i \leq n, e_i \in \mathbb{E}_M(W) \textit{ y es minimal w.r.t. } \preceq_{inf}$$

La Definición 7 determina la estrategia $\mathbb{RDA}$-*strategy*, la cual consiste en reparar todos los errores minimales w.r.t. la relación $\preceq_{inf}$ de un sitio Web. Esto nos lleva a la siguiente proposición, que establece que la $\mathbb{RDA}$-*strategy* permite obtener un sitio Web W libre de errores reparando sólo un subconjunto de los errores de $\mathbb{E}_M(W)$.

Proposition 2. *Sea W un sitio Web y sea $\mathbb{E}_M(W)$ el conjunto de errores de completitud de W. Entonces, (i) la estrategia* $\mathbb{RDA}$-strategy *obtiene un sitio Web libre de errores; (ii) la cantidad de acciones reparadoras ejecutadas por la estrategia* $\mathbb{RDA}$-strategy *es menor o igual al número original de errores en $\mathbb{E}_M$.*

Proof. Sea $e \in \mathbb{E}_M(W)$, dos situaciones son posibles: (i) si e es minimal w.r.t. $\preceq_{inf}$, e es reparado con la operación *repairByDelete*; (ii) si e no es minimal w.r.t. $\preceq_{inf}$, entonces existe un minimal $e' \in \mathbb{E}_M$ s.t. $e' \preceq_{inf} e$ que será reparado y, por la Proposición 1, e será reparado de manera automática sin necesidad de ejecutar una acción reparadora para e.

6.2 Estrategia *reduce-insertion-actions*

El procedimiento *repairByInsert* (descrito en el Algoritmo 1), nos permite reducir la cantidad de información que se debe añadir, como así también la cantidad de acciones reparadoras de inserción a ser ejecutadas en un sitio Web.

Ahora observemos la siguiente situación, donde un error pertenece a más de un conjunto de errores definidos por la relación $\preceq^{sup}$. Sean $\alpha = \{e_i\}_{i=1}^{n}$ y $\beta = \{e'_j\}_{j=1}^{m}$ dos subconjuntos de errores de $\mathbb{E}_M(W)$, tal que $e_1 \preceq^{sup} \ldots \preceq^{sup} e_n$ y $e'_1 \preceq^{sup} \ldots \preceq^{sup} e'_m$, y sea sea e un error de completitud s.t. $e \in \alpha$ y $e \in \beta$. Es claro ver que un error puede pertenecer a más de un conjunto de errores definidos por la relación $\preceq^{sup}$.

Llamaremos $\mathbb{C}_{\mathbb{E}_M}$ a la secuencia de conjuntos de errores formados por la relación $\preceq^{sup}$ como sigue

$$\begin{aligned}\mathbb{C}_{\mathbb{E}_M}(\mathbb{E}_M, \preceq^{sup}) = [c_1, \ldots, c_n] \\ \text{s.t. } (\forall e \in \mathbb{E}_M,\ \exists i, 1 \leq i \leq n, \text{ s.t. } e \in c_i), \\ (\forall i, 1 \leq i \leq n,\ \forall e_1, e_2 \in c_i,\ e_1 \preceq^{sup} e_2 \text{ o } e_2 \preceq^{sup} e_1) \text{ y} \\ (\forall i, 1 \leq i \leq n,\ |c_i| \geqslant |c_{i+1}|)\end{aligned}$$

La secuencia $\mathbb{C}_{\mathbb{E}_M}$ está ordenada por la cardinalidad de los conjuntos que la componen. Denotaremos por $\mathbb{C}_{\mathbb{E}_M}(i)$ a la secuencia $[c_1, \ldots, c_i]$.

De esta manera, podemos definir una partición sobre el conjunto $\mathbb{E}_M$ de la siguiente forma

$$\begin{aligned}\Gamma(\mathbb{E}_M) = \{m_i \mid m_i = dif(\mathbb{C}_{\mathbb{E}_M}(i)),\ \forall i, 1 \leq i \leq k,\ k = |\mathbb{C}_{\mathbb{E}_M}|\},\\ \text{donde}\ \ dif([x_0]) = x_0\\ dif([x_0, \ldots, x_n]) = x_n \setminus \ldots \setminus x_0,\ \text{si}\ n > 0\end{aligned}$$

Definition 8 (estrategia *reduce-insertion-actions*). *Sea W un sitio Web y sea $\mathbb{E}_M(W)$ el conjunto de errores de completitud de W. Una estrategia para reducir la información a insertar (o* $\mathbb{RIA}$-strategy*) para W es*

$$\begin{aligned}[(m_1, repairByInsert), \ldots, (m_n, repairByInsert)],\\ donde\ \forall i, 1 \leq i \leq n,\ m_i \in \Gamma(\mathbb{E}_M(W))\end{aligned}$$

La estrategia de la Definición 8 ejecuta el procedimiento $repairByInsert$ en cada conjunto de la partición de $\mathbb{E}_M$.

Proposition 3. *Sea W un sitio Web y sea $\mathbb{E}_M(W)$ el conjunto de errores de completitud de W y sea $\Gamma(\mathbb{E}_M)$ una partición sobre $\mathbb{E}_M$. Entonces, (i) la estrategia* $\mathbb{RIA}$-strategy *obtiene un sitio Web libre de errores; (ii) la cantidad de acciones reparadoras ejecutadas por la estrategia* $\mathbb{RIA}$-strategy *es menor o igual al número original de errores en $\mathbb{E}_M$.*

Proof. Los conjuntos de la partición $\Gamma(\mathbb{E}_M)$ están ordenados w.r.t. $\preceq^{sup}$ y, como vimos en el Algoritmo 1, la ejecución del procedimiento $repairByInsert$ reduce la cantidad de información necesaria a añadir. De esta manera, con la ejecución de $\mathbb{RIA}$-*strategy* reparamos todos los errores de completitud del sitio Web.

A continuación se presenta un ejemplo para clarificar la utilización de las estrategias *reduce-delete-actions* y *reduce-insertion-actions*.

Example 1. Sea el sitio Web W formado por el conjunto de páginas $\{p_1, p_2, p_3, p_4\}$; y la especificación Web (I_N, I_M, R) con $I_M = \{r_1, r_2, r_3, r_4\}$, de la siguiente forma

Sitio Web $W = \{p_1, p_2, p_3, p_4\}$	Especifcación Web (I_M, I_N, R)
$p_1 = m(s(b), f(a))$	$r_1 = f(X) \rightharpoonup \sharp g(X)\langle A\rangle$
$p_2 = m(m(g(a)))$	$r_2 = g(X) \rightharpoonup \sharp h(X)\langle E\rangle$
$p_3 = m(l(b, a))$	$r_3 = h(X) \rightharpoonup \sharp p(X)\langle A\rangle$
$p_4 = h(b)$	$r_4 = l(X, Y) \rightharpoonup \sharp p(X, Y)\langle A\rangle$

El conjunto de errores de completitud E_M detectados por el proceso de verificación es: E_M = $\{e_1, e_2, e_3, e_4, e_5, e_6\}$, en donde

$e_1 = ((g(a) \rightharpoonup h(a)), \{p_4\}, E)$	$e_4 = ((f(a) \rightharpoonup g(a) \rightharpoonup h(a)), \{p_4\}, A)$
$e_2 = ((h(b) \rightharpoonup p(b)), \{\}, M)$	$e_5 = ((g(a) \rightharpoonup h(a) \rightharpoonup p(a)), \{\}, M)$
$e_3 = ((l(b, a) \rightharpoonup p(b, a)), \{\}, M)$	$e_6 = ((f(a) \rightharpoonup g(a) \rightharpoonup h(a) \rightharpoonup p(a)), \{\}, M)$

Note que e_2, e_3, e_5 y e_6 corresponden a errores de *missing page*, mientras que e_1 es un error existencial y e_4 un error universal.

Los órdenes parciales $\preceq_{inf}$ y $\preceq^{sup}$, de las Definiciones 4 y 5 respectivamente, establecen los siguientes subconjuntos de errores

$$\begin{aligned}&\preceq_{inf} : \{e_1 \preceq_{inf} e_5\}; \{e_2\}; \{e_3\}; \{e_4 \preceq_{inf} e_6\}\\ &\preceq^{sup} : \{e_4 \preceq^{sup} e_1\}; \{e_2 \preceq^{sup} e_3\}; \{e_5 \preceq^{sup} e_6 \preceq^{sup} e_3\}\end{aligned}$$

A continuación describiremos cómo aplicar las estrategias *reduce-delete-actions* y *reduce-insertion-actions* definidas en las secciones 6.1 y 6.2 respectivamente.

Estrategia *reduce-delete-actions*. Se aplica la operación $repairByDelete$ a cada error *minimal* w.r.t. $\preceq_{inf}$:

$$W' = repairByDelete(e_4, repairByDelete(e_3, repairByDelete(e_2, repairByDelete(e_1, W))))$$

donde $W' = \{p_1, p_2, p_3\}$
$p_1 = m(s(b))$
$p_2 = m(m(\ ))$
$p_3 = m(\ )$
$-p_4$ fue eliminada$-$

Estrategia *reduce-insertion-actions*. Para aplicar esta estrategia se siguen tres pasos:

- *Paso 1.* Obtener la secuencia $\mathbb{C}_{\mathbb{E}_M}(\mathbb{E}_M, \preceq^{sup})$

$$\mathbb{C}_{\mathbb{E}_M} = [c_1, c_2, c_3] = [\{e_5, e_6, e_3\}, \{e_2, e_3\}, \{e_4, e_1\}]$$

- *Paso 2.* Realizar la partición $\Gamma(\mathbb{E}_M)$

$$\Gamma(\mathbb{E}_M) = \{m_1, m_2, m_3\}$$
donde $m_1 = c_1 = \{e_5, e_6, e_3\}$,
$m_2 = c_2 \setminus c_1 = \{e_2\}$ y
$m_3 = c_3 \setminus c_2 \setminus c_1 = \{e_4, e_1\}$

- *Paso 3.* Aplicar el procedimiento $repairByInsert$ (descrito en el Algoritmo 1) a cada conjunto $m \in \Gamma(\mathbb{E}_M)$:

$$W' = repairByInsert(m_3, repairByInsert(m_2, repairByInsert(m_1, W)))$$

donde $W' = \{p_1, p_2, p_3, p_4, p_5, p_6\}$
$p_1 = m(s(b), f(a))$
$p_2 = m(m(g(a)))$
$p_3 = m(l(b, a))$
$p_4 = h(b, a)$
$p_5 = p(b, a)$
$p_6 = p(b)$

7 Conclusiones

Encontrarse con información incompleta o indeseada en la Web es muy frecuente, por eso la administración y mantenimiento de la Web es un problema abierto y urgente. En este trabajo, completamos la metodología de reparación semiautomática de sitios Web erróneos presentada en [3] con el tratamiento y eliminación de los errores de completitud. Para ello, establecemos dos relaciones de órden parcial ($\preceq_{inf}$ y $\preceq^{sup}$) sobre los errores de completitud detectados. Con estos órdenes, realizamos un análisis entre la dependencia existente entre los errores, lo cual permitió definir dos estrategias de reparación: *reduce-delete-actions* y *reduce-insertion-actions*. La primera focaliza la reducción de la cantidad de información que es necesaria eliminar para obtener un sitio Web libre de errores con respecto a su especificación. La segunda minimiza la información que es necesaria insertar. Ambas estrategias explotan la dependencia existente

entre los errores, reparando todos sin más que actuar explícitamente sobre un subconjunto del total de ellos.

La consideración de estas estrategias nos lleva a una optimización efectiva del proceso de reaparación de un sitio Web; esto es debido a que el usuario debe de reparar una cantidad menor de errores para obtener un sitio Web correcto y completo. Como trabajo futuro, pensamos en la integración de estas estrategias de reparación en nuestro sistema WebVerdi-M [4] (disponible en http://www.dsic.upv.es/users/elp/webverdi-m/). También hemos comenzado a explorar la posibilidad de utilizar la interpretación abstracta para reducir el coste de la verificación y la reparación.

References

1. S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web. From Relations to Semistructured Data and XML*. Morgan Kaufmann, 2000.
2. M. Alpuente, D. Ballis, and M. Falaschi. Rule-based Verification of Web Sites. *Software Tools for Technology Transfer*, 8:565–585, 2006.
3. M. Alpuente, D. Ballis, M. Falaschi, and D. Romero. A Semi-automatic Methodology for Reparing Faulty Web Sites. In *Proc. of the 4th IEEE Int'l Conference on Software Engineering and Formal Methods(SEFM'06)*, pages 31–40. IEEE Computer Society Press, 2006.
4. M. Alpuente, D. Ballis, M. Falaschi P. Ojeda, and D. Romero. A Fast Algebraic Web Verification Service. In *Proc. of First Int'l Conf. on Web Reasoning and Rule Systems (RR 2007)*, 2007. to appear.
5. F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
6. D. Ballis. *Rule-based Software Verification and Correction*. PhD thesis, University of Udine and Technical University of Valencia, 2005.
7. D. Ballis and D. Romero. Fixing web sites using correction strategies. In *Proc of 2nd Int'l Workshop on Automated Specification and Verification of Web Systems (WWV'06). Paphos, Cyprus*, pages 11–19. IEEE Computer Society Press, 2007.
8. L. Bertossi and J. Pinto. Specifying Active Rules for Database Maintenance. In G. Saake, K. Schwarz, and C. Türker, editors, *Transactions and Database Dynamics, 8th Int'l Workshop on Foundations of Models and Languages for Data and Objects*, volume 1773 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 1999.
9. M. Bezem. *TeReSe, Term Rewriting Systems*, chapter Mathematical background (Appendix A). Cambridge University Press, 2003.
10. F. Bry and S. Schaffert. Towards a Declarative Query and Transformation Language for XML and Semistructured Data: Simulation Unification. In *Proc. of the Int'l Conference on Logic Programming (ICLP'02)*, volume 2401 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.
11. L. Capra, W. Emmerich, A. Finkelstein, and C. Nentwich. XLINKIT: a Consistency Checking and Smart Link Generation Service. *ACM Transactions on Internet Technology*, 2(2):151–185, 2002.
12. N. Dershowitz and D. Plaisted. Rewriting. *Handbook of Automated Reasoning*, 1:535–610, 2001.
13. M. F. Fernandez and D. Suciu. Optimizing Regular Path Expressions Using Graph Schemas. In *Proc. of Int'l Conf on Data Engineering (ICDE'98)*, pages 14–23, 1998.
14. J.W. Klop. Term Rewriting Systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume I, pages 1–112. Oxford University Press, 1992.
15. M. Leuschel. Homeomorphic Embedding for Online Termination of Symbolic Methods. In T. Æ. Mogensen, D. A. Schmidt, and I. H. Sudborough, editors, *The Essence of Computation*, volume 2566 of *LNCS*, pages 379–403. Springer, 2002.
16. E. Mayol and E. Teniente. A Survey of Current Methods for Integrity Constraint Maintenance and View Updating. In *Proc. of Advances in Conceptual Modeling: ER '99*, volume 1727 of *Lecture Notes in Computer Science*, pages 62–73. Springer, 1999.
17. C. Nentwich, W. Emmerich, and A. Finkelstein. Consistency Management with Repair Actions. In *Proc. of the 25th International Conference on Software Engineering (ICSE'03)*. IEEE Computer Society, 2003.
18. Jan Scheffczyk, , Uwe M. Borghoff Peter Rödig, and Lothar Schmitz. S-dags: Towards efficient document repair generation. In *Proc. 2nd Int. Conf. on Computing, Communications and Control Technologies*, volume 2, pages 308–313, 2004.
19. Jan Scheffczyk, Uwe M. Borghoff, Peter Rödig, and Lothar Schmitz. Consistent document engineering: formalizing type-safe consistency rules for heterogeneous repositories. In *Proc. of the 2003 ACM Symposium on Document Engineering (DocEng '03)*, pages 140–149. ACM Press, 2003.
20. Jan Scheffczyk, Peter Rödig, Uwe M. Borghoff, and Lothar Schmitz. Managing inconsistent repositories via prioritized repairs. In *Proc. of the 2004 ACM Symposium on Document Engineering (DocEng '04)*, pages 137–146. ACM Press, 2004.

Gestión de Preferencias de Requerimientos basada en Técnicas Cognitivas

Nadina Martínez Carod and Alejandra Cechich
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue, Neuquén, Argentina
{namartin,acechich}@uncoma.edu.ar

Abstract

Defining software requirements depends on, among other things, the identification, specification, construction and analysis of goals, which capture needs and objectives of the system under consideration. Goal-Oriented Analysis Methods are based on goal refining and goal decomposition. Goals at higher level are customer needs, and they refine until they reach software requirements. Goal-Oriented Analysis (GORA) Methods result in AND-OR goal graphs where each node is a goal. The graph also shows different abstraction levels needed for requirements comprehension. The goal analysis involves conflict resolution among stakeholders, which implies a negotiation process. Our work focuses on stakeholders' preferences as key aspects of negotiation, whereas we use cognitive informatics as negotiation support. In this paper, we introduce an extension of GORA graphs by capture cognitive values that might help reach commitment among stakeholders.

Keywords: Software Requirements Prioritization, Goal Oriented Analysis, Cognitive Informatics, Requirements Elicitation

Resumen

La definición de requerimientos de software depende en gran parte de la identificación, especificación, construcción y análisis de objetivos. Los objetivos capturan las metas y necesidades del sistema bajo consideración. Los métodos de análisis orientados por objetivos se basan en la descomposición y el refinamiento de objetivos. Parten de necesidades y deseos de clientes hasta la obtención de los requerimientos de software. En general resultan en grafos de objetivos AND-OR donde cada nodo es un objetivo. Estos grafos van mostrando además distintos niveles de abstracción necesarios para la real comprensión de los requerimientos. El análisis de objetivos involucra, a su vez, la resolución de conflictos emergentes entre stakeholders, lo que implica un proceso de negociación. Nuestro trabajo está enfocado en las preferencias de los stakeholders como aspectos fundamentales de la negociación, para lo cual utilizamos la informática cognitiva como soporte a dicha negociación. En este artículo, extendemos los grafos de objetivos incluyendo valores cognitivos que podrían ser de ayuda en el proceso de negociación de requerimientos.

Palabras claves: Prioridades en Requerimientos de Software, Análisis Orientado a Objetivos, Informática Cognitiva, Elicitación de Requerimientos

1. Introducción

En los procesos de elicitación de requerimientos, los stakeholders (personas involucradas en un proceso de elicitación como usuarios, analistas, desarrolladores y clientes) tienen diferentes expectativas con respecto al sistema a desarrollar, ya sea por opiniones diferentes o por conflicto de intereses. Esto es debido a que basados en un mismo criterio, cada stakeholder da a cada concepto

un valor preferencial diferente, de acuerdo a sus necesidades y conocimientos. En particular, y considerando requerimientos de un sistema o desarrollo, las diferencias entre stakeholders llevan a conflictos que deben subsanarse. Los conflictos suelen producirse [15] por diferencias tanto en interpretaciones de requerimientos (por ausencias, inconsistencias o diferencias) como en consideraciones sobre la importancia de cada requerimiento.

Las inconsistencias entre requerimientos se producen cuando un stakeholder considera que un requerimiento debe existir y otro stakeholder considera lo contrario; o cuando existe un requerimiento que se contrapone con otro considerado por algún stakeholder. En estos casos es indispensable resolver estas inconsistencias antes de modelar el problema. Cuando dos o más stakeholders interpretan de diferente manera un mismo requerimiento existe lo que se denomina *discordancia de interpretación*. En estos casos el analista debe trabajar en forma grupal, permitiendo el intercambio de información entre stakeholders tratando que todas las personas den el mismo sentido semántico a cada ítem.

Existen situaciones en las que dos o más stakeholders, con una interpretación similar de un mismo requerimiento, lo evalúan diferente. En estos casos hay que mediar para obtener una evaluación en común que mejor satisfaga a los stakeholders con más incidencia en dicho requerimiento.

Para controlar las inconsistencias de requerimientos existen diferentes alternativas: por ejemplo el método GBRAM (Goal-Based Requirements Analysis Method) [1] analiza obstáculos para evitar inconsistencias; en el marco de trabajo KAOS (método formal y riguroso basado en Inteligencia Artificial) [29][4] se utilizan las restricciones de obstáculos para satisfacción de objetivos; y el enfoque propuesto por [23] realiza un análisis de correlación para descubrir incidencias positivas o negativas de objetivos. Para cubrir el caso de los requerimientos discordantes de interpretación se utilizan metodologías de negociación [25][9][12] que consideran acuerdos entre los stakeholders. El primer paso en estas metodologías es mantener un idioma común, lo que requiere que todos los ítems tengan el mismo significado para el total de los stakeholders. El proceso de negociación involucra además priorizar los requerimientos y seleccionar el conjunto de requerimientos a satisfacer.

En los casos de ausencia de requerimientos, el trabajo del analista es ayudar, lo máximo posible, a que los stakeholders detecten los puntos faltantes. En muchos casos la visualización de los requerimientos en metodologías con gráficos o diagramas ayuda a detectar requerimientos que no fueron considerados [13][18][5]. También los métodos como [1][16][17][23], que involucran la descomposición de objetivos, facilitan a los stakeholders a detectar fácilmente requerimientos ausentes al mostrar cómo se va realizando la descomposición.

La evaluación de los requerimientos conlleva un proceso de definición de prioridades entre los mismos. El proceso es bastante complejo, ya que se va complicando al incrementarse la cantidad de requerimientos del sistema a desarrollar. Se pueden utilizar distintas técnicas para definir prioridades en los requerimientos [18][26]. En [20], hemos presentado un marco comparativo, en particular evaluando once de esas técnicas. Para resolver las diferencias en las evaluaciones de los requerimientos, se puede trabajar con una figura mediadora, como el analista, que será el que decide la evaluación final a partir de diferentes evaluaciones; o bien se puede arribar a una evaluación global que se obtenga a partir de las evaluaciones hechas por cada uno de los stakeholders. Esta última opción es hacia la cual nos enfocamos en este trabajo.

Nuestra propuesta intenta mejorar el proceso de evaluación de requerimientos mediante el agregado de factores cognitivos al proceso de elicitación y asignación de prioridades. La Ingeniería Cognitiva es un enfoque interdisciplinario que recurre a disciplinas de la Psicología Cognitiva y de Ciencias de la Computación, a efectos de desarrollar principios, métodos, herramientas y técnicas que guíen el desarrollo de sistemas computarizados para soportar el rendimiento humano. Para esto en

particular tiene en cuenta funciones cognitivas, tales como la resolución de problemas, razonamiento, toma de decisiones, atención, percepción y memoria [24]. Algunos trabajos relacionados hacen uso de técnicas cognitivas para la mejora de procesos de la ingeniería de software; por ejemplo, la propuesta en [22] usa estilos cognitivos como un mecanismo para la selección de personas abocadas a tareas de inspección de software; o la propuesta en [2] usa estilos cognitivos para mejorar la selección de herramientas de trabajo en grupo (groupware) durante el proceso de elicitación de requerimientos distribuido. Nuestro trabajo consiste en aplicar parte de la Psicología Cognitiva – los llamados *estilos cognitivos* – durante el proceso de elicitación de requerimientos y a partir de métodos orientados a objetivos con el fin de mejorar el grado de consenso durante el proceso de priorización de requerimientos.

Para ello, partimos del modelado de objetivos (goals), que son mecanismos lógicos para identificar, organizar y justificar requerimientos de software [1]. Los objetivos son necesarios ya que definen con suficiente completitud una especificación de requerimientos, además son excelentes en la detección de conflictos entre los stakeholders facilitando la forma de resolverlos. Nuestra propuesta extiende el modelado de objetivos incorporando aspectos cognitivos a la evaluación y priorización de los mismos.

Este trabajo está diagramado de la siguiente manera: la sección 2 introduce brevemente los estilos cognitivos y explica las características de nuestra técnica para resolución de conflictos, incluyendo una breve referencia al modelado de objetivos. La sección 3 explica la forma en que se administra la gestión de preferencias en dicha técnica. La sección 4 presenta un caso de estudio. Finalmente, se resumen conclusiones y trabajo futuro.

2. Una Estrategia Cognitiva para la Resolución de Conflictos

2.1 Estilos Cognitivos

La informática cognitiva se describe en [27] como el estudio de lo cognitivo, relacionando las ciencias cognitivas y la informática [31][32] de dos maneras: utilizando técnicas de computación para investigar problemas cognitivos como memoria, aprendizaje, pensamiento y utilizando teorías cognitivas para investigar problemas de ingeniería de Software. Parte de la psicología cognitiva son los estilos cognitivos, basados en la teoría de Jung [14] la cual clasifica las preferencias de las personas sobre percepciones, juicios y procesamiento de la información. Para entender los diferentes modelos de estilos, es conveniente recordar que las personas se relacionan de acuerdo a construcciones, las cuales responden a diversos factores como la edad, experiencia, psicodinámica interna, madurez, cognición, etc. Cada individuo tiene un abordaje predominante que utiliza para percibir, entender, y planear sus interacciones. Dado que el mundo presenta continuamente cantidades enormes de información, el sujeto debe utilizar su estilo de selección, el cual constituye su propio punto de vista.

Como los estilos cognitivos, los modelos de estilos de aprendizaje -Learning Style Models (LSM), son una técnica del campo de la psicología y están basados en observaciones. Clasifican las personas de acuerdo a un conjunto de características sobre la manera en que reciben y procesan la información y son utilizados para mejorar la manera en que las personan aprenden una tarea o labor. Existen diferentes clasificaciones de las estrategias cognitivas que una persona puede emplear en el proceso de elicitación, concernientes a la categorización de los estilos de aprendizaje y en todas ellas se sobreponen las características de unos sobre otros. De esta forma más que considerarse como estilos totalmente definidos, deben considerarse como preferencias o predominios. En [19] se presenta una analogía entre stakeholders y roles en LSMs y consiste en la adaptación de este tipo de modelos a grupos de trabajo en un proceso de elicitación ya que durante el mismo todos los stakeholders "aprenden" uno de otros. Esto es que un stakeholder puede jugar el rol de instructor o

estudiante alternativamente dependiendo de lo que se esté realizando en el momento en que se está llevando a cabo una tarea. Las preferencias por las distintas estrategias de elicitación dependen en buena medida de la forma de ser y de pensar del stakeholder. El objetivo de la investigación de los estilos de aprendizaje es encontrar grupos de personas que utilicen patrones similares de percepción y de interpretación de situaciones. Basándonos en esta información, debemos ser capaces de ajustar los ámbitos del proceso de elicitación para hacerlo más eficiente.

2.2 Características de la Estrategia

Un subconjunto de los métodos orientados a objetivos [1][4][16] [17] desarrolla, como parte del método, un grafo de descomposición de objetivos. En general estos métodos parten de objetivos iniciales, que podrían ser las necesidades de los clientes. Cada objetivo es descompuesto generalmente, en objetivos más específicos (sub-objetivos). Los sub-objetivos son conectados con sus objetivos padres con líneas con direccionamiento. Existen dos tipos de descomposición: La descomposición AND, donde un objetivo se satisface si todos los sub-objetivos correspondientes a ese objetivo se satisfacen. Y la descomposición OR, cuando al menos un sub-objetivo se satisface el objetivo padre se satisface. La descomposición de objetivos puede continuar en varios niveles diferentes de abstracción creando una jerarquía. Existen muchas situaciones, incluso en que un sub-objetivo podría ser parte de más de un objetivo.

Esta estrategia complementa cualquier técnica de elicitación orientada por objetivos tomando como inicio un único grafo AND-OR resultante de la aplicación de la técnica orientada por objetivos utilizada. Partiendo de dicho grafo, el analista junto con cada stakeholder discuten el grafo global, modificándolo y creando una vista parcial de dicho grafo de acuerdo a las apreciaciones del stakeholder. Luego, a partir de las modificaciones establecidas en los grafos parciales, se ajusta el grafo global y se verifica el nivel de disidencia entre las vistas parciales y la global de acuerdo a los objetivos generales y finales. En caso de que el valor de disidencia sea elevado se vuelve a realizar todo el análisis, guardando en un software de soporte las objeciones de los expertos a los valores asignados. Este proceso es iterativo hasta llegar a un mínimo de acuerdo entre los stakeholders.

Nuestro método extiende el grafo AND-OR de objetivos agregándole un valor a la percepción de los stakeholders con respecto al objetivo en sí mismo y con respecto a la técnica de elicitación utilizada en la obtención de cada objetivo. Así, el grafo global del sistema estará compuesto por un grafo de objetivos, donde cada objetivo tiene adjunto la técnica de elicitación utilizada en el mismo o la considerada más apropiada para utilizar. Aquellos objetivos que no tienen proceso de elicitación, se considerarán con la misma técnica que la de su objetivo-padre. Si tienen más de un objetivo padre, o si difieren en algo al objetivo padre, este subobjetivo deberá tener adosado la técnica de elicitación utilizada. Cada stakeholder tiene una visión parcial donde se deberán mostrar además aquellos objetivos que los stakeholders no están autorizados a definir, con una explicación de la razón por la cual no está autorizado.

Dado que nuestra meta es definir el grado de preferencia de las personas por ciertas técnicas con diferentes características de trabajo, analizamos sólo las técnicas con mayor utilización en el proceso de elicitación, según Hickey & Davis [11] (ver Figura 1). Esto puede ser ampliado utilizando cualquier técnica de elicitación no considerada por el momento.

Partimos de una lista predefinida de objetivos del sistema, que son las necesidades de los usuarios, y de una lista de stakeholders con información de los mismos. De acuerdo a las preferencias de los stakeholders obtendremos la tendencia de cada stakeholder con respecto a las técnicas de elicitación. Llegamos así a una nueva ponderación que llamaremos carga cognitiva la cual determina el grado de "satisfacción" del stakeholder por la técnica que se utilizó para elicitar tales objetivos.

Sin embargo, por más que el stakeholder se sienta "a gusto" con la técnica de adquisición del requerimiento, si éste no ha comprendido suficientemente lo que se busca con dicho objetivo, su opinión no va a tener el mismo peso que la de otro individuo que sí lo haya comprendido y analizado. Entonces, incorporamos una nueva valuación que será el *valor cognitivo*; valor comprendido entre 0 y 1, que representa el grado de acuerdo de un stakeholder sobre un objetivo determinado y depende de su perfil cognitivo – representando el grado de comprensión alcanzado debido a su perfil relacionado con las características de la/s técnica/s de elicitación.

En resumen, el peso cognitivo asignado por cada stakeholder a cada subobjetivo estará dado por el perfil cognitivo del stakeholder en relación a la técnica de elicitación utilizada en ese objetivo o subobjetivo.

La siguiente figura muestra como ejemplo la vista parcial del proceso de elicitación para un cierto stakeholder, donde las elipses representan cada uno de los objetivos, los rectángulos son las técnicas utilizadas o posibles de utilizar por cada uno de los objetivos y los valores son las valuaciones asignadas por ese stakeholder para cada objetivo. Vemos que el objetivo *I* y *J* heredan la técnica de elicitación adosada al objetivo *G* que es prototipación. En la misma figura vemos que el objetivo *D* tiene más de una técnica de elicitación (casos de uso y lluvia de ideas). El objetivo *E* no debe ser evaluado por este stakeholder y la razón está especificada en el llamado de dicho nodo. El objetivo *H* tiene adosada una matriz de preferencia [15][16] con los valores que están recuadrados.

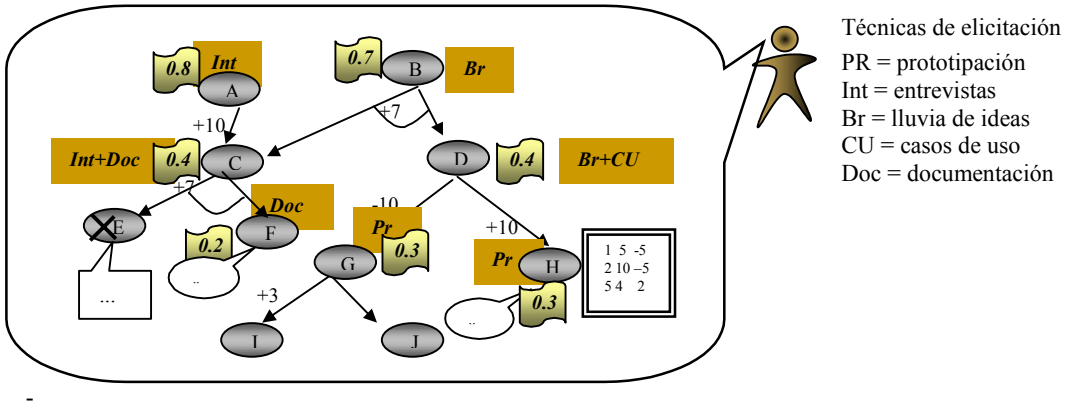


Figura 1 -Grafo de objetivos extendido

La persona que juega el rol del facilitador es la que corroborará el estado con los valores asignados por cada stakeholder, indicando el valor resultante, producto de las características cognitivas de cada stakeholder en cada uno de los objetivos de acuerdo a la/s técnica/s de elicitación utilizada/s. Ahora, visto desde el punto de vista de un objetivo, su valor estará compuesto por las valuaciones de todos los stakeholders. Una descripción más detallada de la técnica, incluyendo su descripción formal, puede verse en [21] .

3. Determinación del valor cognitivo

La determinación del valor cognitivo estará influenciada por las preferencias de los stakeholders. Para ello se necesita una administración o gestión de esas preferencias. Luego de analizar varios modelos de aprendizaje, encontramos que el modelo propuesto por Felder-Silverman [19] (F-S) es lo suficientemente general y adoptamos ese modelo en la gestión de preferencias. En el modelo de F-S [7], las personas, de acuerdo a las características de comportamiento [6][8], pueden caer dentro de una categoría u otra dependiendo de las circunstancias, donde las categorías a las cuales nos

vamos a remitir son: perceptivo/intuitivo, visual/verbal, activo/reflexivo, y secuencial/global. Las preferencias pueden ser *fuertes, moderadas o leves*. Sólo cuando una persona tiene una preferencia fuerte, puede ser categorizada dentro de un cierto grupo. Para determinar preferencias dentro de las subcategorías *perceptivo/intuitivo, visual/verbal, activo/reflexivo, secuencial/global* se utiliza un instrumento que es el Índice de Estilos de Aprendizaje (Index Learning Styles). El ILS es un test compuesto por 44 preguntas cuya versión online se encuentra en [28]. Mediante su aplicación, obtendremos las características personales del stakeholder, que nos darán una idea acerca de sus preferencias y de la manera en que dicho stakeholder procesa la información.

La gestión de preferencias se realiza en dos etapas bien diferenciadas. La primera, cuando se obtienen las preferencias predominantes de cada técnica de elicitación teniendo como base el comportamiento de cada persona. Y la segunda, cuando las preferencias se analizan estadísticamente en conjunto con una muestra grande de personas agrupando características similares. Luego, se debe obtener las características de comportamiento de cada stakeholder en función del modelo de Felder-Silverman y a partir de la información de las estadísticas generadas, se pueden obtener las preferencias de esa persona respecto a la técnica de elicitación en particular. La razón por la cual la obtención de la técnica de elicitación apropiada para una persona a partir de estadísticas es más cercana a la realidad que por un cuestionario es que en general los usuarios o clientes ya sea por desconocimiento, mala información, o experiencia negativa, desconocen su preferencia respecto de las técnicas de elicitación. En particular podemos asegurar que el valor cognitivo será más cercano a la realidad cuando mayor sea la muestra para la obtención de preferencias.

Como la negociación debe ser realizada considerando varios stakeholders, el problema radica en cómo se seleccionarán las preferencias de los mismos, o bien a quién se considerará con mayor relevancia para dicha selección. A tal efecto se estudiaron posibles variantes, la primera [10] da mayor preponderancia a las preferencias del analista, [2] la segunda considera más importante las preferencias en común de la mayoría y la tercera considera sólo las preferencias más marcadas, dejando sin efecto las demás preferencias.

Partimos de la función selectora π de Hickey & Davis [10] que se basa en las preferencias del analista. El problema de esta función de selección es que puede suceder que los stakeholders en general se sientan incómodos con esta preferencia, y por ende la comunicación no sea óptima, lo que traería como consecuencia resultados poco satisfactorios. Como ejemplo supongamos que el analista sea, de acuerdo al índice ILS, levemente reflexivo, y fuertemente verbal, por lo que preferiría escuchar opciones, tal vez por medio de entrevistas o lluvia de ideas. Pero si los stakeholders tienen fuerte preferencia visual y son fuertemente activos, las preferencias entre el analista y el segundo grupo se contraponen, con lo que stakeholders y analista no se sentirán cómodos. El resultado no será tan confiable como si la elicitación se hubiera realizado con técnicas más acordes al grupo.

Por ello, extendemos la función selectora, llamada ahora π^*, para realizar un análisis de preferencias de cada stakeholder obteniendo la propuesta más representativa entre los mismos. Si bien esto permite que la generalidad de los stakeholders esté en conformidad con la técnica utilizada, todavía existe la posibilidad de que algunos stakeholders sean parte de una minoría no representada. Surge así la función selectora π^{**}.En este caso puede suceder que la mayoría no tenga una fuerte preferencia, y en cuyo caso, se podrían elegir técnicas con fuerte "rechazo" para un grupo minoritario, en perjuicio del resultado general.

La función π^{**} mejora a la anterior considerando el grado de preferencia de los stakeholders. Esta mejora toma en cuenta las preferencias fuertes dejando sin efecto las preferencias moderadas o leves, lo que optimiza los resultados de las dos funciones previas ya que el grado de comodidad o

incomodidad es mucho más marcado en preferencias fuertes que en moderadas o leves. En estos casos, se priorizan los rechazos o deseos de los stakeholders con fuertes preferencias, de esta manera se pueden suavizar las preferencias no deseadas de stakeholders con preferencias no fuertes.

El análisis se realiza considerando los casos extremos, en los cuales las personas poseen características muy definidas. Los casos extremos serían los que están dentro de las categorías *fuertes* de cualquier subcategoría del modelo F-S. Sobre estos casos se calcula la inclinación por medio de porcentajes y sumatorias: se suman los porcentajes de las personas con fuerte preferencia que consideraron una técnica determinada dentro de las técnicas preferidas y se restan los porcentajes con fuerte preferencia que consideran a dicha técnica como inapropiada, dejando sin considerar los casos para los cuales la técnica es indiferente. La estadística de preferencias genera un valor estipulado para todas las técnicas de elicitación evaluadas de acuerdo a las preferencias fuertes en un grupo de personas.

Al terminar el análisis de acuerdo a las predominancias, tendremos valores para cada una de las técnicas de elicitación. En la Tabla 1 mostramos como ejemplo personas fuertemente activas y visuales con valores determinados para tres técnicas de elicitación.

Predominancia	Técnica de elicitación	Valor
Activo	Casos de uso Documentación Entrevistas	0.53 0.78 0.27
Visual	Casos de uso Documentación Entrevistas	0.83 0.65 0.42

Tabla 1. Valores de preferencias para algunas técnicas de elicitación según perfiles activo/visual

Una vez generadas las estadísticas de preferencias, estas quedarán con valores que podrán ser utilizados por varios proyectos. En el momento que se decida, los valores pueden ser regenerados.

En síntesis, para cada una de las categorías del Modelo F-S se considera sólo las personas con fuerte predominancia en dicha categoría, (lo que equivale a decir fuertemente perceptivo, fuertemente intuitivo, ídem en el caso de visual/verbal, activo/reflexivo, y secuencial/global). De esa cantidad de personas y para una técnica de elicitación en particular se suman la cantidad de personas que la consideraron preferida y se restan las que la consideraron inapropiada.

Dicho de otra manera el valor en cognitivo para las personas con categoría fuerte "*Fc"* correspondiente a una técnica de elicitación "*j"* está dada por la formula:

$$V(FC_c[tec_j]) = \frac{1}{n}\left(\sum_{i=1}^{n} FC_c[+tec_j] - \sum_{i=1}^{n} FC_c[-tec_j]\right),$$

donde $[+tec_j]$ significa preferencia por la técnica de elicitación "*j"* y $[-tec_j]$ significa fuerte rechazo por dicha técnica de elicitación.

4. Caso de estudio

Tomamos como ejemplo el sistema de inscripción en el impuesto de Ingresos Brutos a realizarse en Rentas de la Provincia de Neuquén, Argentina.

Previo a analizar el sistema se realizó la adquisición de valores de preferencia. Primero se analizó una muestra relativamente chica, cuyo resultado fue que para algunas técnicas no se podía obtener el valor preferido al no ser consideras ni preferidas ni inapropiadas por personas con fuerte

preferencia por una categoría. Por lo que se determinó que la muestra debía ser extensa. En particular se prefirió que la determinación de preferencias se realice con perfiles variados en edad, conocimientos técnicos e informáticos, utilizando también personas que trabajan en dominios con organización similar.

El objetivo principal del sistema es "Inscribir una persona física o jurídica en ingresos brutos para ejercer una actividad". Adicionalmente, se debe llevar la documentación requerida, el sistema debe calcular el monto fijo a pagar de acuerdo a la actividad del negocio/servicio, se debe llenar un formulario de acuerdo a los ingresos y se inscribe en ingresos brutos. A partir de allí todos los meses se debe pagar a mes vencido un porcentaje de acuerdo al monto de facturación.

La Figura 2 muestra parte del grafo AND/OR correspondiente a un momento dentro del proceso de elicitación. Es importante destacar en la figura que los objetivos son las elipses. Los rectángulos grises con bordes están compuestos por los vectores de *Valor* (V) y de *Peso Cognitivo* (PC) para todos los stakeholders. En este ejemplo V_a es el vector de valores del nodo *a* para cada uno de los stakeholders. V_a está compuesto por los valores $(v_1, v_2,...., v_{i-1}, ,v_{i+1},...., v_n)$. Lo que significa que todos los stakeholders, salvo el stakeholder i determinaron el valor asignado a dicho objetivo. De la misma manera el PC_a es el vector de pesos cognitivos. Donde $PC_a = (pc_1, pc_2,.., pc_{i-1}, ,pc_{i+1},.., pc_n)$. Ambos vectores son añadidos a los objetivos en el grafo global.

Los otros rectángulos coloreados especifican, por medio de abreviaturas, los métodos de elicitación a utilizar por el analista. En nuestro ejemplo tenemos que el objetivo *e* que es *"verificar documentación"* está considerado con dos técnicas de elicitación, las cuales están separadas por el signo +. Las técnicas de elicitación del ejemplo son *PR* (prototipación), *Int* (entrevistas), *Br* (lluvia de ideas), *CU* (casos de uso), y *Doc* (documentación). Los objetivos que no tienen adosada una técnica de elicitación, como el objetivo *Rapidez,* conservan la adosada a su objetivo padre *(Int)*. Cada uno de los vectores Valor está determinado de acuerdo a los valores establecidos por los stakeholders para un objetivo. El valor del vector V_e, estará compuesto por los valores de todos los stakeholders considerados para el objetivo *Verificar Documentación*. Los vectores de *Peso Cognitivo* se determinan a partir de los valores de preferencia.

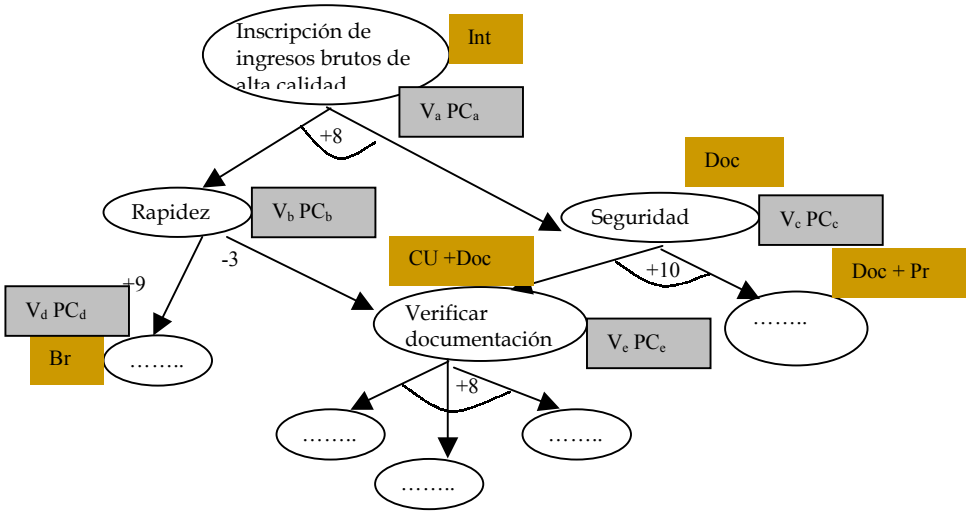


Figura 2. Parte del grafo extendido del caso de estudio "Inscripción a Ingresos Brutos"

¿Cómo se utilizan los valores de preferencia en la determinación de los Pesos Cognitivos?

Sucedieron las siguientes situaciones:

1 La situación más simple fue para la persona p_4 en el objetivo inicial obj_a con los vectores *(Va PCa)*. Dicha persona tiene una fuerte preferencia visual y dicho objetivo fue adquirido

mediante la técnica de elicitación *Int.* En este caso la determinación del valor cognitivo para esa persona fue dada por *V(FVi[Int])*. Considerando la tabla se determina de acuerdo al valor ubicado en la preferencia *visual* para la técnica *Int, entonces*

- *PC (p_4, obj_a) = V(FVi[Int]) = 0.42.* Este valor indicaría el grado de afinidad de p_4 correspondiente a la técnica de elicitación *Int* (considerando 1 como lo óptimo).

2 El caso de p_2, quien tiene fuerte predominancia por más de una categoría (*visual y activa),* para el mismo objetivo obj_a, (adquirido por una única técnica de elicitación). En este caso la persona utilizaría sus características más afines a la técnica realizada, es por ello que se elige el máximo valor entre las dos características para la misma técnica.

- *PC (p_2, obj_a) = Max (V(FAc[Int]), V(FVi[Int]))*
  PC (p_2, obj_a)= Max (0.27, 0.42) = 0.42. Este valor indicaría que frente a dos características de una persona, ésta usará aquella con la cual se sienta más cómoda en una técnica de elicitación (visual en este caso).

3 La tercer situación fue nuevamente para p_4 pero considerando el objetivo obj_e. En estos casos el mismo objetivo se realiza por más de una técnica, donde cada técnica complementa a otra técnica (se detectan situaciones que no se podrían observar de otra manera). Aquí se realiza la media de los valores de todas las técnicas utilizadas. La persona es fuertemente *visual* y el objetivo fue obtenido por *Casos de uso* y *Documentación*.

- *PC (p_4, obj_e) = (V(FVi [CU]) + V(FVi [Doc]))/ 2*
  PC (p_4, obj_e) = (0.83+0.65) / 2 = 0.74. Este valor indicaría que si bien una de las técnicas es más preferida de acuerdo a las características de una persona, al utilizar varias técnicas la información resultante se verá afectada por la preferencia asociada a las otras técnicas – será una media de la suma de las mismas.

4 El caso de p_2 considerando el objetivo obj_e fue la siguiente situación. La persona tiene fuerte predominancia por más de una categoría y el objetivo fue adquirido por más de una técnica de elicitación. En estos casos el valor cognitivo será la media del mayor valor de las técnicas de elicitación que tuvieron intervención, para las predominancias que posea la persona. El objetivo se obtiene por *Casos de Uso y Documentación**,*** y la persona tiene fuerte predominancia *activa y visual*, entonces el peso cognitivo de dicha persona es:

- *PC (p_2, obj_e) = (Max(V(FAc[CU]),V(FVi[CU])) + Max(V(FAc[Doc]),V(FVi [Doc])))/2*
  PC (p_2, obj_e) =(0.83 + 0.78)/2 = 0.805. Este valor indicaría que en cada una de las técnicas de elicitación, la persona utiliza aquellas características de mayor afinidad acorde a la técnica.

Por lo tanto el grafo de objetivos global variará no sólo a las valuaciones de cada stakeholder sino el peso cognitivo, por lo que resultará en un análisis más robusto.

Esta estrategia también puede ser utilizada para elegir qué técnicas podrían ser viables para una buena comunicación entre las partes. En particular se podría realizar ensayos en los que el analista "probara" que incidencia tendría el cambiar una técnica por otra, siempre y cuando la técnica sea posible (no tendría sentido utilizar por ejemplo una técnica de *brainstorming* cuando parte del grupo de trabajo no puede conciliar momentos comunes para la técnica). Si bien lo óptimo para elicitar cada objetivo sería utilizar todos los métodos de elicitación posibles, para mejorar el peso cognitivo asignado por cada persona, esto sería inviable por una cuestión de recursos (tiempo, dinero y

personas). Entonces una vez más hay que encontrar un equilibrio entre las técnicas de elicitación utilizadas y los resultados obtenidos a partir de ellas, es decir hay que considerar el punto justo de costo-beneficio al obtener mejores valores de preferencia cambiando técnicas o agregando nuevas.

Si bien esta metodología está pensada para ser utilizada en todos los objetivos dentro de un grafo de objetivos, para elicitar requerimientos con más criterio, se podría considerar sólo aquellos objetivos en conflicto, en aplicaciones de gran envergadura, cuyas dimensiones excedan las posibilidades de utilizar este método en todos los objetivos. También sirve para, en los casos de tipo OR, considerar la prioridad de los objetivos, para definir qué objetivo implementar primero e incluso no implementar. También se podría definir desarrollar ciertos objetivos en versiones posteriores de un determinado desarrollo. En cualquier caso, si se utilizara lo cognitivo en todos los objetivos, se aseguraría mayor entendimiento de cualquiera de los objetivos, tengan o no tengan conflictos.

5. Conclusiones

Las tareas involucradas en la elicitación de requerimientos tienen un carácter esencialmente colaborativo, y participan en ella varios stakeholders con distintas especialidades y conocimiento heterogéneo. Por tal motivo, es de suponer que aparecerán conflictos entre los stakeholders. Para sobrellevar esta situación planteamos una estrategia de resolución de conflictos que prioriza valuaciones de cada stakeholder de acuerdo a su perfil cognitivo mediante una clasificación del comportamiento de las personas utilizando estilos de aprendizaje. Nos inclinamos por aspectos cognitivos, no desde el punto de vista de la adquisición de conocimiento, sino de las características de los stakeholders. Nuestro trabajo intenta mejorar el proceso de elicitación de requerimientos, a partir de métodos orientados a objetivos, utilizando los mecanismos de información afectan el comportamiento humano.

Nuestra estrategia se basa principalmente en preferencias en técnicas de elicitación y perfiles cognitivos como elementos para solucionar valuaciones diferentes de un mismo requerimiento. La estrategia puede ser aplicada como complemento de técnicas orientada por objetivos, donde la gestión de preferencias nos ayudará a obtener el valor cognitivo de una persona por un determinado objetivo siguiendo alguna posible técnica de elicitación – y consecuentemente mejorará el análisis del grado de consenso obtenido. Sin embargo, es necesario validar más extensamente la propuesta. Para ello, estamos desarrollando herramientas de soporte al cálculo y definiendo casos de estudio usando esas herramientas. Esperamos que ambos resulten en beneficios al proceso de negociación de requerimientos en la práctica.

Agradecimientos

Este trabajo es resultado del proyecto de investigación UNComa 04/E059 (Mejora del Proceso de Desarrollo de Software Basado en Componentes).

Referencias

[1] Antón A. *"Goal Based Requirements Analysis"* In Proceedings of the 2nd International Conference on Requirements Engineering (ICRE '96) IEEE software April 15 - 18, 1996
[2] Aranda G. et al. *"A Cognitive-Based Approach to Improve Distributed Requirements Elicitation Processes"*, In Proceedings of the Second IEEE International Conference on Cognitive Informatics (ICCI´05), 2005.
[3] Boehm B.W., Grünbacher P., Briggs B. *"Developing Groupware for Requirements Negotiation: Lessons Learned"*. IEEE Software, May/June 2001, pp. 46-55.
[4] Dardenne A., van Lamsweerde A., and Fickas S, 1993. *"Goal-directed Requirements Acquisition"*. Science of Computer Programming Vol. 20, pp. 3-50.

[5] Edwards, W. and Barron, F.H., *"SMARTS and SMARTER: Improved Simple Methods for Multiattribute Utility Measurement"*, Organizational Behavior and Human Decision Processes 60, 1994, pp. 306-325.
[6] Felder, R.: *"Matters of Styles"*. ASEE Prism, vol. 6, no. 4, (1996), pp 18-23.
[7] Felder R.M. <available at: http://www.ncsu.edu/felder-public/Papers/LS-Prism.htm>
[8] Felder, R., Silverman, L.: *"Learning and Teaching Styles in Engineering Education"*. Engineering Education, vol. 78, no. 7, (1988, preface 2002), pp 674-681.
[9] Grüenbacher P. "*Collaborative Requirements Negotiation with EasyWinWin"* 2nd International Workshop on the Requirements Engineering Process, Greenwich, London IEEE Computer Society,2000. ISBN 0-7695-0680-1. p9.954-690.
[10] Hickey, A.M. and Davis, A. *"Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes"*. In 36th Annual HawaiiInternational Conference on Systems Sciences (HICSS), January 2003, 96-105.
[11] Hickey, A. and Davis, A, *"Elicitation Technique Selection: How Do Experts Do It"* in Proceedings of the 11th IEEE International Engineering Conference, 2003.
[12] In H., Olson D., Rodgers T. *"A Requirements Negotiation Model Based on Multi-Criteria Analysis"*. Fifth IEEE International Symposium on Requirements Engineering (RE '01). August 27-31, 2001. Toronto, Canada.
[13] In H. and Roy, S., *"Visualization Issues for Software Requirements Negotiation"*, IEEE International Computer Software and Applications Conference (COMPSAC 2001), Chicago, Illinois, USA, pp. 10-15, October 2001.
[14] Jung CG. *"Psychological Types"*. Pantheon Books, London. Internet. 1923.
[15] Kaiya H. et al. *"Improving the detection of requirements discordances among stakeholders"* Requirements Engineering, vol.10, no.4, pp.289–303, Dec. 2005
[16] Kaiya H., Horai H., and Saeki M., *"AGORA: Attributed Goal-Oriented Requirements Analysis Method"*, In Proceedings of the IEEE International Conference on Requirements Engineering, 2002, pp. 13-22.
[17] KAOS homepage, http://www.info.ucl.ac.be/research/projects/AVL/ReqEng.html
[18] Karlsson, J. and Ryan, K. *"A Cost-Value Approach for Prioritizing Requirements"*. IEEE Software, Vol. 14(5): p. 67-74, September/October 1997.
[19] Martín A., Martínez C., Martínez Carod N., Aranda G., and Cechich A. *"Classifying Groupware Tools to Improve Communication in Geographically Distributed Elicitation"*. IX Congreso Argentino en Ciencias de la Computación, CACIC 2003, La Plata, 6-10 Octubre 2003, (942-953).
[20] Martinez Carod, N. and Cechich, A. *"Classifying Software Requirement Prioritization Approaches"*. XI Congreso Argentino en Ciencias de la Computación, CACIC 2005, Entre Ríos, 6-10 Octubre 2005.
[21] Martinez Carod, N and Cechich, A. *"A Cognitive Psychology Approach for Balancing Elicitation Goals"*, Accepted In the Sixth IEEE International Conference on Cognitive Informatics (ICCI´07), California, USA 2007.
[22] Miller, J. and Yin, Z., "A Cognitive-Based Mechanism for Constructing Software Inspection Teams". IEEE Transactions on Software Engineering, 30(11): 2004, 811-825.
[23] Mylopoulos J., Chung L., Liao S., Wang H., Yu E., "*Exploring Alternatives During Requirements Analysis"*, IEEE Software, Jan/Feb 2001, Vol: 18 p 92-96.
[24] Roth E., Patterson E., Mumaw R. *"Cognitive Engineering: Issues in User-Centered System Design"*. J.J. Marciniak (Ed.), Encyclopedia of Software Engineering, 2º Edition. New York: Wiley-Interscience, John Wiley & Sons, 110-123, 1994.
[25] Ruhe G., Ebertein A., Pfahl D. *"Quantitative WinWin – A New Method for Decision Support in Requirements Negotiation"*. SEKE'02, Italy, July 2002. ACM.
[26] Saaty T.L., 1990. *"The Analytic Hierarchy Process"*. McGraw-Hill.
[27] Shi Z., Shi J.. *"Perspectives On Cognitive Informatics"*. In Proceedings of the Second IEEE International Conference on Cognitive Informatics (ICCI´03), pages 129-137, 2003.
[28] Soloman B. Felder R., *"Index of Learning Styles Questionnaire"* <available at: http://www.engr.ncsu.edu/learningstyles/ilsweb.html>
[29] van Lamsweerde, A *"From system Goals to Software Architecture"*. In Proceedings of the SFM. p. 25--43, 2003.
[30] van Lamsweerde A.*"Goal-Oriented Requirements Engineering: A Guided Tour*". Source Proceedings of the 5th IEEE International Symposium on Requirements Engineering, (2001)
[31] Wang Y. *"Cognitive Informatics: A New Transdisciplinary Reseearch Field"*. (2003)

[32] Wang Y. *"On Cognitive Informatics"*. In Proceedings of the First IEEE International Conference on Cognitive Informatics. (ICCI´02), Calgary, Alberta, Canada, August 2002, pp 34-42

Índice espacio-temporal D*R-Tree y sus avances experimentales

María G. Dorzán, Edilma O. Gagliardi
Universidad Nacional de San Luis, Facultad de Ciencias Físico,
Matemáticas y Naturales, Departamento de Informática
San Luis, Argentina, D5700HHW
{ mgdorzan, oli }@unsl.edu.ar

Resumen
Las Bases de Datos Espacio-Temporales permiten almacenar y consultar los cambios de posición, forma y/o tamaño de objetos a lo largo del tiempo. Los métodos espacio-temporales permiten responder consultas que involucran predicados espacio-temporales en forma eficiente, donde los más considerados, en general, son *TimeSlice, Eventos, Intervalo* y *Trayectoria*. En la literatura, podemos encontrar una gran variedad de métodos, los cuales intentan optimizar el desempeño de las consultas, pero siempre apuntando a un subconjunto de las antes mencionadas.

En este artículo presentamos *D*R-Tree*, un índice espacio-temporal, que permite resolver estos cuatro tipos de consultas, sin aumentar la complejidad espacio-temporal.

Presentamos una evaluación experimental más avanzada, que se realizó considerando aspectos propios de una realidad, y que mostró el buen desempeño del índice en aplicaciones de diferentes magnitudes.

Palabras claves: bases de datos espacio-temporales, índices espacio-temporal, consultas espacio-temporal.

1. INTRODUCCIÓN

Actualmente han surgido nuevas aplicaciones que requieren una eficiente manipulación de objetos móviles y de las relaciones existentes entre ellos. Por consiguiente, los sistemas de base de datos deberían ser capaces de soportar estos nuevos tipos de datos y de contar con lenguajes de consultas apropiados. Así, para la recuperación de información, podrían considerarse métodos de acceso espacio-temporales, que surjan como estructuras adicionales o bien índices que resulten de extensiones de los métodos de acceso multidimensional, satisfaciendo de esta manera, las demandas de los usuarios [4, 5, 6, 8].

En este sentido, nuestro trabajo consistió en desarrollar un método de indexación espacio-temporal, D*R-Tree, con el objetivo de evaluar y diseñar estructuras de datos y algoritmos que permitan resolver eficientemente los principales tipos de consultas espacio-temporales tales como *TimeSlice*, *Intervalo*, *Eventos* y *Trayectoria* [1]. Para su diseño y desarrollo, nos basamos en el modelo propuesto en [2], llamado SEST-Index, dado que aprovechamos sus ventajas respecto de una buena utilización de espacio en disco y de su buen tiempo para responder consultas, como así también de la posibilidad que nos permitió ampliar el conjunto de consultas espacio-temporales.

En este artículo presentamos avances experimentales realizados sobre D*R-Tree. La evaluación experimental la realizamos en dos etapas. En la primera, la evaluación consistió en respetar las condiciones de implementación de SEST-Index con el fin de tener los mismos parámetros de comparación. En la segunda etapa, consideramos una evaluación experimental que tuviera más relación con aplicaciones reales, de modo que los parámetros de referencia caracterizaran mejor el comportamiento del índice en situaciones de mayor magnitud. De este modo, la evaluación experimental consolidó la eficiencia del método, a través de los resultados obtenidos, los cuales fueron satisfactorios, mostrando el buen desempeño del índice al responder un conjunto de consultas espacio-temporales, más amplio que el resto de los métodos.

El presente artículo está organizado como sigue: en la sección 2 describimos las estructuras que constituyen el índice D*R-Tree y presentamos los algoritmos de consultas para soportar el procesamiento de las mismas. En la sección 3 mostramos los resultados obtenidos en las dos etapas de la evaluación experimental. Finalmente, damos nuestras conclusiones, y el marco y visión de futuro de nuestro trabajo.

Este trabajo está subvencionado por el Proyecto Tecnologías Avanzadas de Bases de Datos 22/F314, Departamento de Informática, Universidad Nacional de San Luis, y por el Proyecto AL07-PAC-027 Geometría Computacional, de la Universidad Politécnica de Madrid, España.

2. D*R-TREE

D*R-Tree es un método de acceso espacio-temporal que se basa en la idea expuesta en [2]. Sus estructuras permiten almacenar y recuperar objetos espacio-temporales de forma eficiente. Este método está diseñado para responder consultas de tipo TimeSlice, Eventos, Intervalo y Trayectoria. Trata de mantener un equilibrio entre el espacio de disco utilizado por la estructura y el tiempo de acceso empleado en responder los distintos tipos de consulta. En la Figura 1 se muestra el esquema del método de acceso espacio-temporal.

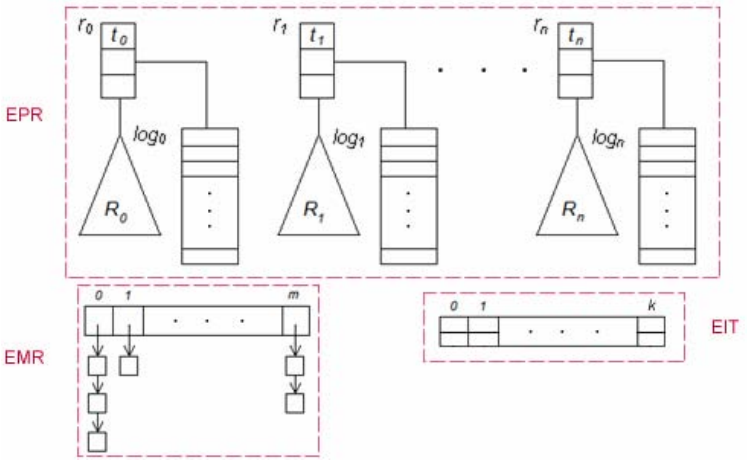


Figura 1: Esquema de D*R-Tree

2.1 ESTRUCTURAS

***Estructura Puntos de Referencia* (*EPR*)**

Almacena instantes de tiempo específicos, llamados puntos de referencia: r_0, r_1,…, r_n. Cada instante de tiempo tiene asociado un *snapshot* de la base de datos. Los puntos de referencia se almacenan en una lista secuencial ordenada por tiempo. Las entradas son tuplas de la forma $r_i = < t_i, R\text{-}root_i, L\text{-}head_i >$, donde t_i corresponde al instante de tiempo asociado al punto de referencia; $R\text{-}root_i$ mantiene la raíz del *i*-ésimo R-Tree [3]; y $L\text{-}head_i$ mantiene la cabecera de la *i*-ésima bitácora. Por tanto, esta estructura tiene asociada una *Colección de R-Trees* (*CRT*) y una *Colección de Bitácoras* (*CB*). Cada snapshot se almacena en un R-Tree, R_i, el cual está asociado a un punto de referencia r_i. Los cambios que ocurren entre puntos de referencia consecutivos, r_i y r_{i+1}, se almacenan en una bitácora, log_i, asociada a un punto de referencia r_i. Cada bitácora es una lista secuencial ordenada por tiempo. Las entradas son tuplas de la forma <*Oid, t*, *Back-log*, *Mbr-act*> donde *Oid* es el identificador del objeto; *t* es el instante de tiempo asociado al movimiento ocurrido; *Back-log* mantiene un puntero a una entrada de una bitácora que almacena el movimiento previo del objeto con identificador *Oid*; y *Mbr-act* mantiene la última posición espacial almacenada del objeto.

***Estructura de Movimientos de Referencia* (*EMR*)**

Esta estructura es una lista secuencial de listas, cuyas entradas del primer nivel poseen la siguiente forma: <*Oid*, *List-head*> donde *Oid* es el identificador del objeto y *List-head* es la cabecera de la lista vinculada l_i. Cada nodo de la lista vinculada l_i tiene la siguiente forma: < *Log-ref*, *Rp-time*, *Next-n*>, donde *Log-ref* es un puntero a la entrada de una bitácora donde se almacena el último movimiento del objeto con identificador *Oid*; *Rp-time* es el instante de tiempo asociado al punto de referencia donde se mantiene la bitácora indicada por *Log-ref*; y *Next-n* es el puntero al siguiente nodo de la lista. Para cada objeto o_i se mantiene un acceso al último movimiento de o_i almacenado en cada bitácora. Se debe tener en cuenta que existe, como máximo, un nodo asociado a cada bitácora y una bitácora está asociada, como máximo, a un solo nodo.

Estructura de Índice de Tiempo (EIT)

Para cada instante de tiempo t_i se mantiene un acceso al primer movimiento registrado en el instante t_i en la bitácora correspondiente. Los instantes de tiempo son almacenados en una lista secuencial ordenada por tiempo. Esta estructura permite encontrar cualquier instante de tiempo en diferentes bitácoras. Las entradas son tuplas de la siguiente forma < *t*; *Log-ref* >, donde *t* es un instante de tiempo y *Log-ref* es un puntero a la entrada de una bitácora donde se almacena el primer movimiento ocurrido en el instante de tiempo *t*.

2.2 CONSULTAS

El tipo de *Consultas de selección* es de la forma "*encontrar todos los objetos que se encontraban en un área o punto específico, durante un intervalo o instante de tiempo específico*". Las consultas típicas son: i) *TimeSlice*: responde consultas expresadas de la siguiente forma: "*encontrar los objetos que se encontraban en un área específica en un instante de tiempo dado*"; ii) *Intervalo*: responde consultas expresadas de la siguiente forma: "*encontrar los objetos que se encontraban en un área específica en un intervalo de tiempo dado*". El tipo de consulta *Eventos* es de la forma: "*recuperar los eventos que ocurrieron en una región en un instante de tiempo dado*". La consulta específicamente requiere recuperar los objetos que entraron o salieron de alguna región de consulta en un instante de tiempo dado. Finalmente, el tipo de consulta *Trayectoria* es de la forma: "*recuperar un conjunto de posiciones espaciales por las cuales ha pasado un objeto en un intervalo de tiempo dado*" [6, 7, 9]. A continuación mostramos los algoritmos diseñados para responder los tipos de consulta descriptos anteriormente:

Algoritmo *TimeSlice* (*R, t*)
Entrada: *R* es el rectángulo de consulta y *t* es el instante sobre el cual se realiza la consulta.
Salida: *Q* es el conjunto de objetos que satisfacen la consulta.
1. $i \leftarrow FindTime(t)$
2. $Q \leftarrow FindInRTree(EPR(i).R\text{-}root, R)$
3. While $EPR(i).t \leq t$ do
4. $\quad Q \leftarrow UpdateQwithLog(EPR(i).L\text{-}head, R, t)$
5. endWhile
6. Return *Q*

Algoritmo *Intervalo* (R, t_o, t_f)
Entrada: *R* es el rectángulo de consulta, t_o es el límite inferior del intervalo de tiempo y t_f es el límite superior sobre el cual se realiza la consulta.
Salida: *Q* es el conjunto de objetos que satisfacen la consulta.
1. $i \leftarrow FindTime(t_o)$
2. $Q \leftarrow FindInRTree(EPR(i).R\text{-}root, R)$
3. While $EPR(i).t \leq t_o$ do
4. $\quad Q \leftarrow UpdateQwithLog(EPR(i).L\text{-}head, R, t_o)$
5. endWhile
6. If todas las entradas de la bitácora $EPR(i).L\text{-}head$ fueron procesadas then
7. $\quad i \leftarrow i + 1$
8. endIf
9. While $EPR(i).t \leq t_f$ do
10. $\quad Q \leftarrow UpdateQwithLog(EPR(i).L\text{-}head, R, t_f)$
11. $\quad i \leftarrow i + 1$
12. endWhile
13. Return *Q*

Algoritmo *Eventos* (*R*, *t*)
Entrada: *R* es el rectángulo de consulta y *t* es el instante sobre el cual se realiza la consulta.

Salida: *Q* es el conjunto de objetos que satisfacen la consulta.
1. $E \leftarrow FindInLogByT(EIT, t)$
2. For each $e \in E$ do
3. If *e.Mbr_act* está en $R \wedge$ *e.Back-log* no está en R then
4. $Q \leftarrow Q \cup \{e\}$ /* entró a R*/
5. else
6. If *e.Mbr_act* no está en $R \wedge$ *e.Back-log* está en *R* then
7. $Q \leftarrow Q \cup \{e\}$ /* salió de *R* */
8. endIf
9. endIf
10. endFor
11. Return *Q*

Algoritmo *Trayectoria* (*Oid*, t_i, t_k)
Entrada: *Oid* es el identificador del objeto a consultar, t_i es el límite inferior del intervalo de tiempo y t_k es el límite superior sobre el cual se realiza la consulta.
Salida: *Q* es el conjunto de objetos que satisfacen la consulta.
1. $c \leftarrow EMR(Oid).List\text{-}head$
2. While $t_k \leq c.Rp\text{-}time$ do
3. $c \leftarrow c.Next\text{-}n$
4. endWhile
5. $b \leftarrow c.Log\text{-}ref$
6. $Q \leftarrow Q \cup \{b.MBR\_act\}$
7. $b \leftarrow b.Back\text{-}log$
8. While $(b.t > t_i) \wedge (b \neq -1)$ do /*si b = -1, se está referenciando al primer R-Tree*/
9. $Q \leftarrow Q \cup \{b.MBR\_act\}$
10. $b \leftarrow b.Back\text{-}log$
11. endWhile
12. If $b = -1$ then
13. $Q \leftarrow FindInicialPosition(Oid)$
14. endIf
15. Return *Q*

Donde:
- *FindTime*(*t*) mediante búsqueda binaria, encuentra el punto de referencia que corresponde a la posición en *EPR* tal que *EPR*(i).t ≤ t.
- *FindInRTree*(*EPR*(*i*).*R-root*, *R*) encuentra los objetos almacenados en el R-Tree con raíz *EPR*(*i*).*R-root* que intersecan *R*.
- *UpdateQwithLog*(*EPR*(*i*).*L-head*, *R*, *t*) actualiza el conjunto de resultado *Q* con los objetos almacenados en la bitácora referenciada por *EPR*(*i*).*L-head*. Los instantes de tiempo de dichos objetos deben ser menor o igual a *t* y su MBR debe intersecar a la región *R*.
- *FindInLogByT*(*EIT*, *t*) utilizando la estructura *EIT* encuentra las entradas de la bitácora que almacena todos los movimientos asociados al instante de tiempo *t*.
- *FindInicialPosition*(*Oid*) encuentra el MBR inicial del objeto con identificador *Oid*.

3. EVALUACIÓN EXPERIMENTAL

*D*R-Tree s*e implementó en lenguaje *Java* utilizando el ambiente de desarrollo Eclipse v3.0.1. Las pruebas se realizaron en una computadora *AMD Athlon 64* de 1.8 Ghz con 1Gb. de RAM, en el sistema operativo *Windows*. Los lotes de prueba se obtuvieron utilizando el generador de datos espacio-temporales GSTD[10]. En las diferentes pruebas se tuvieron en cuenta las siguientes condiciones: la cantidad de objetos en movimiento es fija; el porcentaje de movilidad es alto; los objetos informan los cambios de posición; los objetos se mueven en el plano, en una región acotada y conocida; se utilizan distribuciones de probabilidades para generar los movimientos de los objetos; y el área de las consultas son rectángulos cuyos lados son perpendiculares a los ejes. Cabe aclarar que las pruebas se realizaron para aquellos casos donde la cantidad de entradas necesarias

para almacenar los movimientos producidos en un instante de tiempo no sobrepasa la capacidad de la bitácora. Por tal motivo, algunos experimentos no se muestran ya que no pueden brindar información relevante. La utilización del disco se midió por el número de bloques utilizados por la estructura de datos luego de haber insertado los objetos y sus respectivos movimientos. El tiempo de acceso se definió como el número promedio de bloques leídos luego de haber realizado 100 consultas aleatorias.

3.1 PRIMERA ETAPA

Fue realizada considerando un universo de objetos moviéndose en el plano, con los mismos lotes de prueba utilizados en la evaluación experimental del método base [2]. En esta etapa se respetaron las condiciones de implementación de SEST-Index, evaluando la misma cantidad de objetos, de bloques por bitácora, entre otros. A continuación presentamos los resultados más relevantes; para mayores detalles que los expuestos ver [1]. En las comparaciones se consideró la utilización de almacenamiento y tiempo de acceso a disco en las consultas de tipo TimeSlice, Intervalo, Eventos y Trayectoria.

Esta etapa consistió en evaluar los distintos algoritmos con lotes de prueba de 1000, 3000 y 5000 objetos. Los objetos fueron modelados como puntos en movimiento. Estos puntos fueron distribuidos uniformemente dentro de una región en el instante de tiempo 0.0. Seguidamente, fueron movidos aleatoriamente durante los siguientes 50 instantes hasta alcanzar el instante de tiempo 1.0. Se utilizaron los siguientes porcentajes de movilidad (frecuencia de cambio): 1%, 3%, 5%, 7%, 9%, 11%, 13% y 15% por instante de tiempo. Esto es, que por instante de tiempo se mueve un porcentaje de la cantidad de objetos considerados. Además, se utilizaron bitácoras de 4, 8 y 12 bloques por bitácora con bloques de disco de 1024 bytes.

3.1.1 *Utilización del espacio*

Podemos observar que para ambas estructuras, a medida que aumenta el porcentaje de movilidad mayor es la cantidad de bloques necesarios para almacenarlas. También, observamos que la diferencia en la cantidad de bloques utilizados por ambas estructuras aumenta a medida que se incrementa el porcentaje de movilidad.

La experimentación nos mostró que al aumentar la cantidad de objetos, la cantidad de bloques utilizados aumentó proporcionalmente. También que, mientras la cantidad de bloques por bitácora aumenta, menor es el espacio utilizado por la estructura en general ya que se necesitan mantener menos puntos de referencia. La Figura 2 muestra la cantidad de bloques utilizados en mantener la estructura en ambos métodos con lotes de 5000 objetos, 12 bloques por bitácora y porcentaje de movilidad de 1%, 3%, 5%, 7%, 9% y 11%. La diferencia en el uso del disco, se debe a que nuestra propuesta responde eficientemente a la consulta de tipo Trayectoria, lo que implica el uso de estructuras adicionales y modificaciones en la bitácora. Sin embargo, este costo adicional en el uso de espacio representa un porcentaje bajo adicional respecto de la estructura original, que obviamente resulta poco significativo en comparación a las ventajas obtenidas.

3.1.2 *Consultas*

Para todos los tipos de consulta se realizaron 100 consultas aleatorias. Para el caso de TimeSlice, Intervalo, Evento y Trayectoria se utilizaron rectángulos que cubren un 5% y 10% del área total y la posición del rectángulo y el instante de tiempo de la consulta son aleatorios. Además, para las consultas de tipo Intervalo se utilizaron intervalos de tiempo de tamaño 5 y 10 unidades.

Luego de realizar la experimentación notamos que en D*R-Tree no hay significativas variaciones en el número de bloques leídos, para responder los tipos consulta TimeSlice, Intervalo y Evento con

respecto a SEST-Index ya que ambos índices trabajan de forma similar para resolverlos. Las Figuras 3, 4 y 5 muestran la cantidad de bloques leídos por ambos métodos para responder a los tipos de consulta TimeSlice, Intervalo y Evento con lotes de 5000 objetos, 12 bloques por bitácora, porcentaje de movilidad de 1%, 3%, 5%, 7%, 9% y 11%, y áreas del 5%. Para la consulta de tipo Intervalo se muestran los resultados obtenidos con tamaño de intervalo de 10 unidades.

Cabe aclarar que como *SEST-Index* no responde al tipo de consulta Trayectoria en forma eficiente por no estar diseñada para este fin, se debe encontrar el punto de referencia que contiene el instante correspondiente al límite superior del intervalo de consulta y luego recorrer secuencialmente las bitácoras, en forma descendiente con respecto al tiempo, hasta encontrar el instante correspondiente al límite inferior del intervalo. Como respuesta a la consulta Trayectoria se retornan las posiciones encontradas a lo largo de este recorrido que corresponden al objeto de consulta. *D*R-Tree* simplemente utiliza la estructura adicional *EMR* para encontrar el último movimiento del objeto de consulta en la bitácora correspondiente al límite superior del intervalo. Luego, se recorren las bitácoras utilizando los punteros a una posición de una bitácora en la cual está almacenado el movimiento inmediatamente anterior correspondiente objeto de la consulta. De esta manera, se asegura que sólo se leen los bloques donde existe una referencia a un movimiento del objeto. Además notamos que *SEST-Index* puede no retornar un resultado ya que en intervalos de tiempo donde no se han registrado movimientos del objeto, en el recorrido secuencial no se encontrará información del objeto de consulta. También, se realizaron pruebas donde se compara D*R-Tree con distintos intervalos de tiempo de consulta y se observó que al incrementar el intervalo de consulta, la diferencia entre la cantidad de bloques leídos no es significativa. Los resultados que se muestran en la Figura 6 corresponden a pruebas realizadas para 5000 objetos, con intervalo de 10 unidades de tiempo con 12 bloques por bitácora.

3.2 SEGUNDA ETAPA

La segunda etapa fue motivada por una aplicación real en la cual los parámetros anteriores resultaban limitados. Por lo tanto, la evaluación experimental consistió en utilizar lotes de prueba de 10000 y 20000 objetos. Estos objetos, modelados como puntos moviéndose en el plano, al igual que en la primera etapa, fueron distribuidos uniformemente dentro de una región comenzando en el instante de tiempo 0.0. Estos objetos continuaron moviéndose durante los siguientes 200 instantes hasta alcanzar el instante de tiempo 1.0. Los porcentajes de movilidad utilizados fueron de 1%, 3% y 5%. Notar que al aumentar el porcentaje de movilidad, se incrementa sustancialmente la cantidad de objetos que se mueven en un instante de tiempo. Por tanto, también se requieren bitácoras más grandes. Los tamaños de bitácora usados fueron de 8, 16, 32 y 64 bloques por bitácora con tamaño de bloque de disco de 1024 bytes. En esta segunda etapa las pruebas se realizaron sobre D*R-Tree.

La utilización del disco se midió por el número de bloques utilizados por la estructura de datos luego de haber insertado los objetos y sus respectivos movimientos. El tiempo de acceso se definió como el número promedio de bloques leídos luego de haber realizado 100 consultas aleatorias.

3.2.1 *Utilización del espacio*

Para esta etapa, hicimos la evaluación experimental sólo sobre D*R-tree, observando cuál era el crecimiento de la utilización del espacio en disco. En el caso en que la experimentación consistió en mover 10000 objetos, con porcentajes de movilidad: 1%, 3% y 5%, podemos observar en la Figura 7 en el caso del 1%, cuando se consideran bitácoras de 4 bloques, ocurre que el espacio en disco utilizado es alrededor de los 12500 bloques. Esto se debe a que las bitácoras son pequeñas y se deben determinar más puntos de referencia y en consecuencia existen más árboles R-trees. El caso del 3%, se inicia con bitácoras de 16 bloques, para poder registrar al menos todos los movimientos de un instante de tiempo asociado a un punto de referencia. Como es lo mínimo indispensable para

el tamaño de una bitácora, resulta ser que la cantidad de bloques utilizados también es elevada, superando los 25000 bloques usados. También aquí la justificación es similar.

Para el caso del 5% de movilidad, se requiere iniciar con bitácoras de al menos 32 bloques, con la misma justificación: un instante completo debe poder guardarse en ella; y la cantidad de bloques usados en total alcanza los 20000. Notemos que cuando las bitácoras contuvieron más bloques, entonces los puntos de referencia comenzaron a alejarse temporalmente, lo cual permitió tener árboles RTrees más distanciados en el tiempo y bitácoras que eventualmente almacenaran los movimientos correspondientes a varias instancias. Así, cuando las bitácoras son de 64 bloques, podemos observar que la cantidad de bloques utilizados se reduce sustancialmente, indicando que el número adecuado en estos casos ronda en bitácoras de dicho tamaño.

En el caso en que la experimentación consistió en mantener 20000 objetos, con porcentajes de movilidad: 1% y 3% podemos observar la Figura 8. En el caso del 1%, cuando se consideran bitácoras de 4 bloques, ocurre que el espacio en disco utilizado es alrededor de los 12500 bloques, tal como se vio en el caso anterior. También, se debe a que las bitácoras son pequeñas y se deben determinar más puntos de referencia y en consecuencia hay más árboles R-Trees. El caso del 3%, la cantidad de bloques asciende a 22000, con bitácoras de 64 bloques, ya que es lo mínimo necesario. Claramente, se puede observar, con casos anteriores, con bitácoras más grandes, se reduce significativamente la cantidad de bloques utilizados.

3.2.2 *Consultas*

Para la consulta TimeSlice, la Figura 9 nos muestra para 10000 objetos con porcentajes de movilidad 1%, 3%, 5% y 7%, la cantidad de bloques leídos, donde el área de la consulta es del 5% del área total. Al igual que en la primera etapa de la evaluación experimental, esta consulta tiene un comportamiento heterogéneo en el sentido de cómo se determinan las curvas. Podemos observar que para el 1% de movilidad, con bitácoras de 16 bloques se necesitan leer menos de 15 bloques, para el 3% alrededor de 16 o 17 bloques y crece más en el caso del 5%, donde se leen cerca de 24 bloques. Nuevamente, tiene que ver con el tamaño de las bitácoras, puesto que requiere buscar en bitácoras el instante de tiempo dado y los puntos de referencia están más distanciados.

En el caso del 3%, se da una coincidencia entre bitácoras de 16 y 32 bloques porque en ambos casos se almacenan la misma cantidad de instantes de tiempo, habiendo una con más espacio desperdiciado (porque llega el nuevo punto de referencia y hay que construir el R-Tree asociado, dejando atrás la bitácora). En el caso del 3% con bitácoras de 64 bloques, se observa el crecimiento de la cantidad de bloques leídos, ya que almacena varios instantes de tiempo. Cuando el porcentaje de movilidad es 5%, con bitácoras de 64 bloques, baja la cantidad de bloques leídos respecto del caso anterior, y esto se debe a que la bitácora almacena menos instantes de tiempo. En los casos del 1% y 3%, se mantiene lo mismo. En el caso de movilidad del 7%, con bitácoras de 64 bloques, ocurre similar al caso del 3% de movilidad con bitácoras de 64 bloques.

Dado los resultados de la consulta TimeSlice (Figura 10), para 20000 objetos con porcentajes de movilidad 1% y 3%, donde el área de la consulta es del 5% del área total, la cantidad de bloques leídos disminuye. Lo interesante de este caso, es que con una población de 10000 objetos, con porcentajes de movilidad similares, se mantuvo estable respecto de la cantidad de bloques leídos.

Para la consulta Intervalo, podemos observar que mientras más grande es el porcentaje de movilidad, mayor es el número promedio de bloques leídos, ya que para cada instante de tiempo se necesitan almacenar mayor cantidad de entradas en la bitácora. Por la misma razón, para mayor cantidad de objetos, mayor es la cantidad de bloques leídos. Si bien el tamaño de la bitácora influye en el espacio utilizado para almacenar la estructura como mencionamos anteriormente, no afecta en

la cantidad de bloques leídos para responder este tipo de consulta. Notamos que no hay variaciones significativas entre los resultados obtenidos para 10000 y 20000 objetos.

Respecto de Eventos, observamos que mientras más grande es el porcentaje de movilidad, mayor es el número promedio de bloques leídos, ya que en cada instante de tiempo ocurren más eventos. La cantidad de bloques por bitácora no disminuye el rendimiento de la estructura. Cabe aclarar que el área de consulta no influye en los resultados ya que en todos los casos, debemos examinar todos los movimientos que se produjeron en el instante consultado.

Para Trayectoria, notamos que el porcentaje de movilidad y la cantidad de objetos no afecta el desempeño del algoritmo que responde a este tipo de consulta ya que el diseño de la estructura permite acceder sólo a aquellas posiciones donde el objeto ha registrado un movimiento.

Como muestran las siguientes figuras, se mantiene casi estable la cantidad de bloques leídos, aunque el porcentaje de movilidad y la cantidad de objetos varíen.

CONCLUSIONES

En este artículo presentamos D*R-Tree, un índice espacio-temporal, que integra la resolución de cuatro tipos de consultas espacio-temporales, sin aumentar la complejidad espacio-temporal. Presentamos su estructura, los algoritmos de consulta y la evaluación experimental realizada, la que mostró el buen desempeño del índice en aplicaciones de diferentes magnitudes respecto de la población de objetos en movimiento.

Durante las dos etapas de la evaluación experimental, notamos que una variable importante en la evaluación experimental de D*R-Tree es el tamaño de la bitácora, ya que éste determina no sólo que tan grande puede llegar a ser el espacio utilizado por la estructura, sino también el tiempo de acceso que se utilizará para responder las consultas. Para determinar el tamaño de bitácora adecuado que se ajuste mejor a nuestras necesidades, debemos tener en cuenta que a medida que el tamaño de la bitácora aumenta, menor será la utilización del espacio. Esto se debe a que la estructura en general mantendrá menos R-Trees.

Sin embargo, el hecho de tener bitácoras más grandes fuerza a aumentar la cantidad de búsquedas secuenciales dentro de las bitácoras, deteriorando consultas del tipo TimeSlice e Intervalo ya que en ambos casos, la cantidad promedio de bloques leídos sería mayor.

El valor indicado para las dimensiones de la bitácora tiene que tener en cuenta ambos aspectos.

Concluimos que nuestra estructura resulta una contribución al área de investigación, ya que proponemos un método de acceso espacio-temporal que responde eficientemente los principales tipos de consulta: TimeSlice, Intervalo, Eventos y Trayectoria.

ANEXO FIGURAS

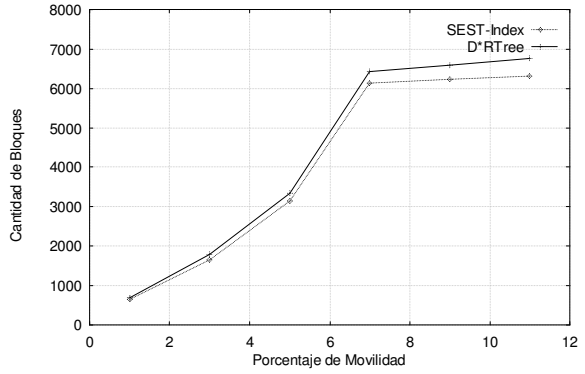


Figura 2: Número de bloques utilizados para 5000 objetos (12

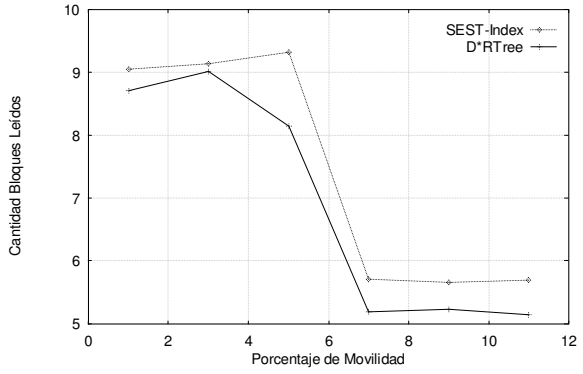


Figura 3: Número de bloques leídos para TimeSlice en un

bloques por bitácora)

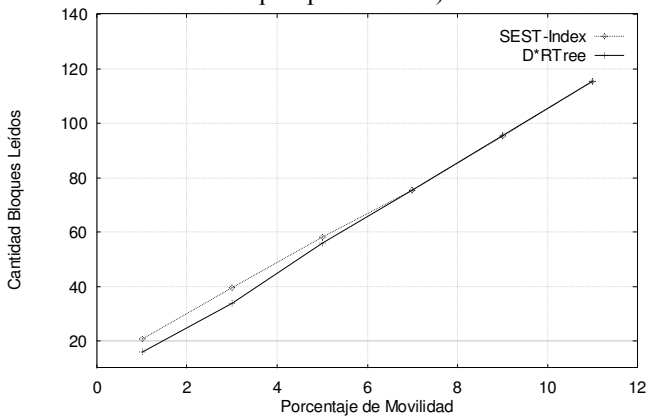


Figura 4: Número de bloques leídos para Intervalo de tamaño 10 en un área de 5% para 5000 objetos (12 bloques por bitácora)

área de 5% para 5000 objetos (12 bloques por bitácora)

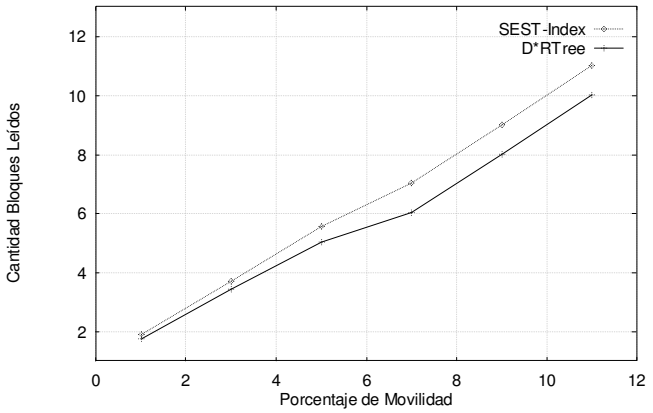


Figura 5: Número de bloques leídos para Eventos en un área de 5% para 5000 objetos (12 bloques por bitácora)

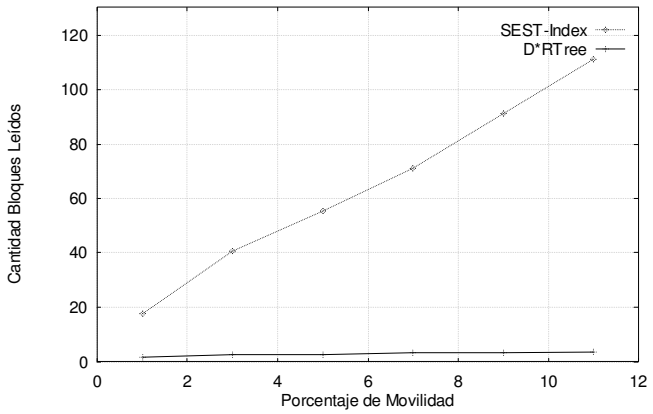


Figura 6: Número de bloques leídos para Trayectoria con intervalo de 10 unidades de tiempo para 5000 objetos (12 bloques por bitácora)

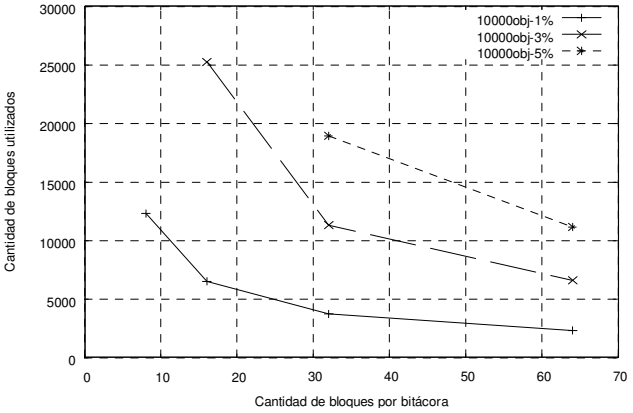


Figura 7: Número de bloques utilizados para 10000 objetos

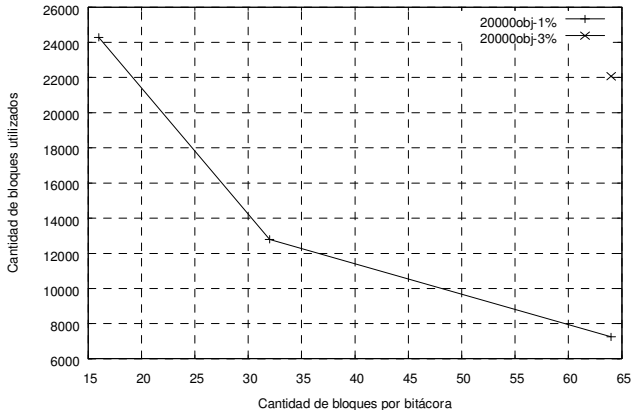


Figura 8: Número de bloques utilizados para 20000 objetos

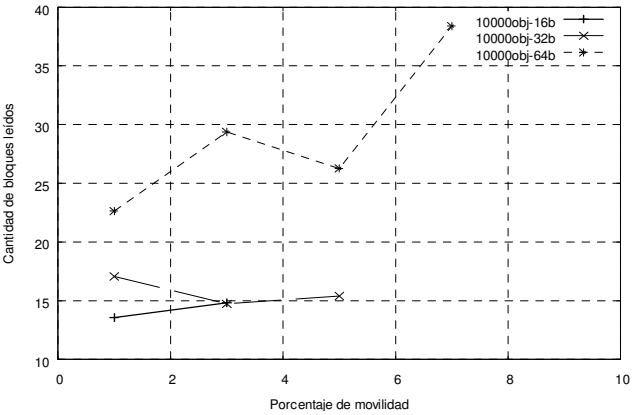


Figura 9: Número de bloques leídos para TimeSlice en un área de 5% para 10000 objetos

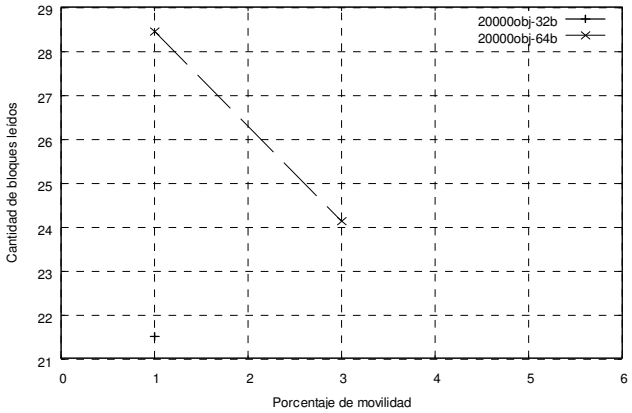


Figura 10: Número de bloques leídos para TimeSlice en un área de 5% para 20000 objetos

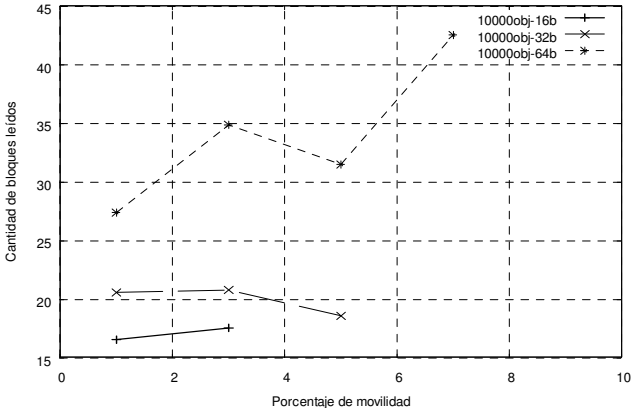


Figura 11: Número de bloques leídos para TimeSlice en un área de 10% para 10000 objetos

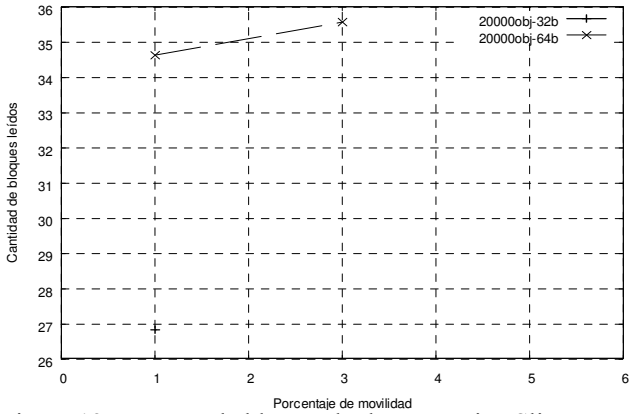


Figura 12: Número de bloques leídos para TimeSlice en un área de 10% para 20000 objetos

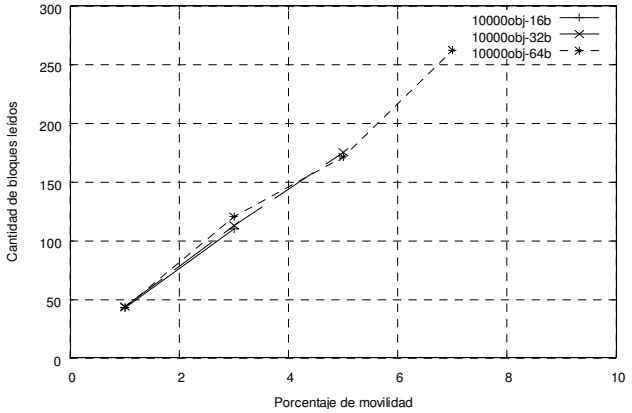


Figura 13: Número de bloques leídos para Intervalo en un área de 5% para intervalos de 5 unidades para 10000 objetos

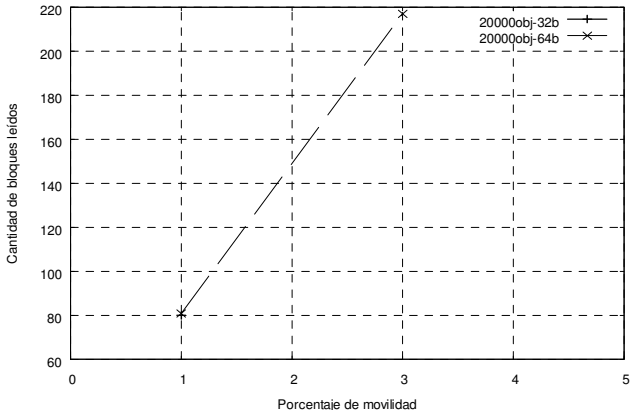


Figura 14: Número de bloques leídos para Intervalo en un área de 5% para intervalos de 5 unidades para 20000 objetos

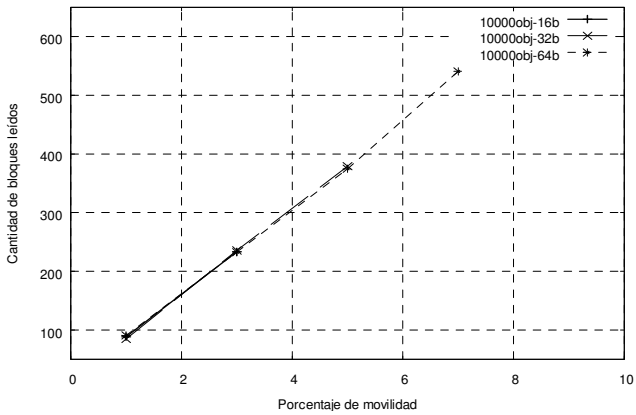


Figura 15: Número de bloques leídos para Intervalo en un área de 10% para intervalos de 10 unidades para 10000 objetos

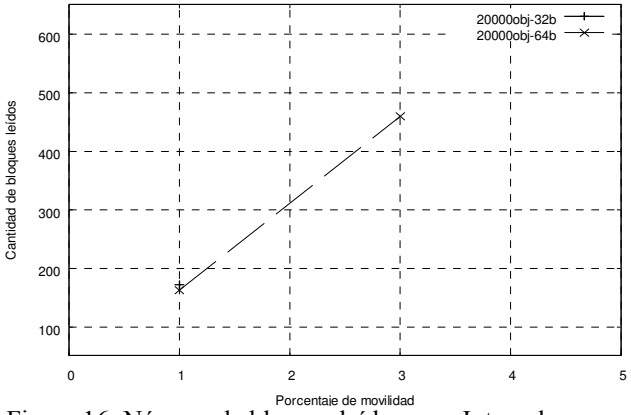


Figura 16: Número de bloques leídos para Intervalo en un área de 10% para intervalos de 10 unidades para 20000 objetos

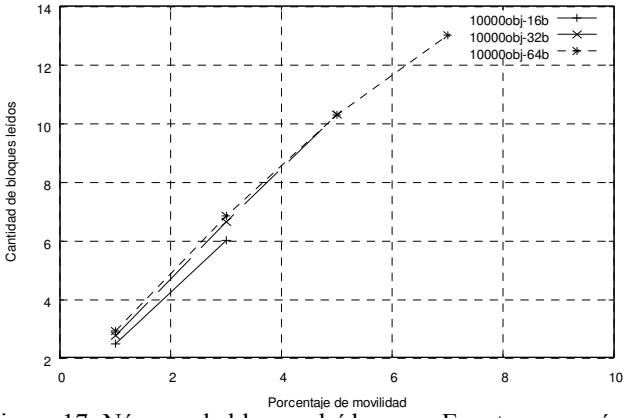


Figura 17: Número de bloques leídos para Eventos en un área de 5% para 10000 objetos

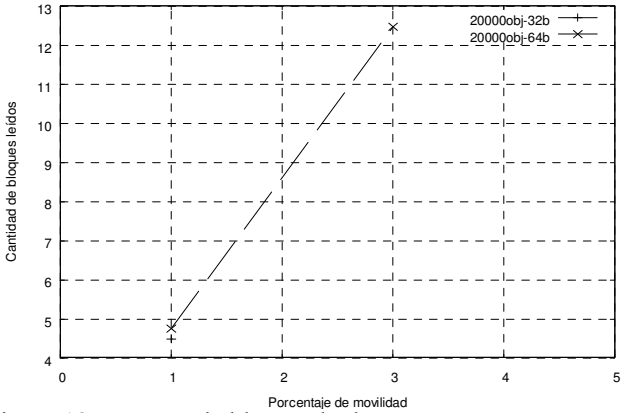


Figura 18: Número de bloques leídos para Eventos en un área de 5% para 20000 objetos

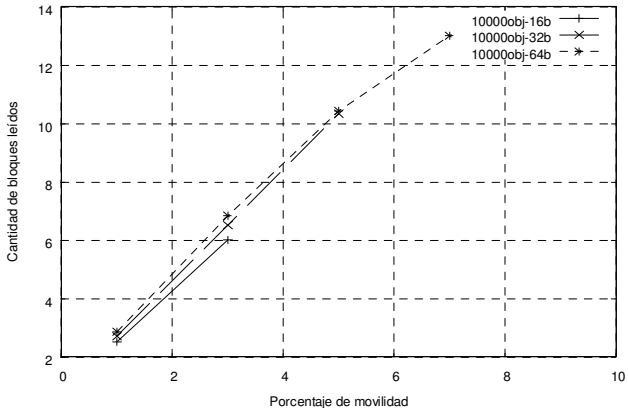


Figura 19: Número de bloques leídos para Eventos en un área de 10% para 10000 objetos

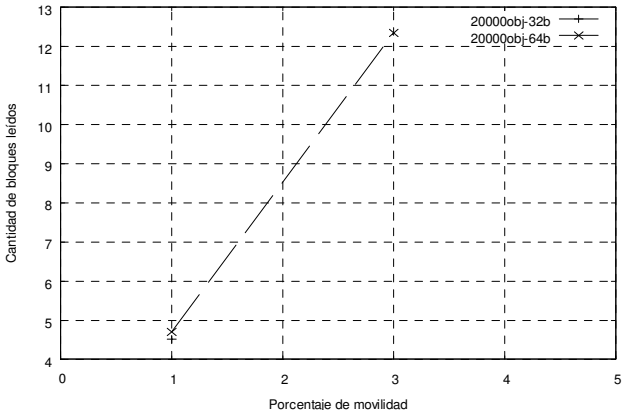


Figura 20: Número de bloques leídos para Eventos en un área de 10% para 20000 objetos

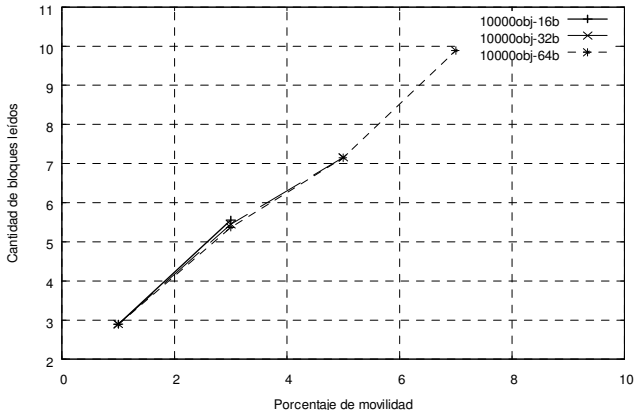


Figura 21: Número de bloques leídos para Trayectoria en un área de 5% para intervalos de 5 unidades para 10000 objetos

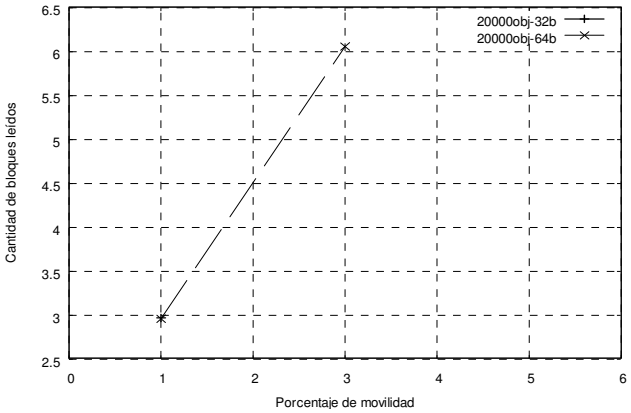


Figura 22: Número de bloques leídos para Trayectoria en un área de 5% para intervalos de 5 unidades para 20000 objetos

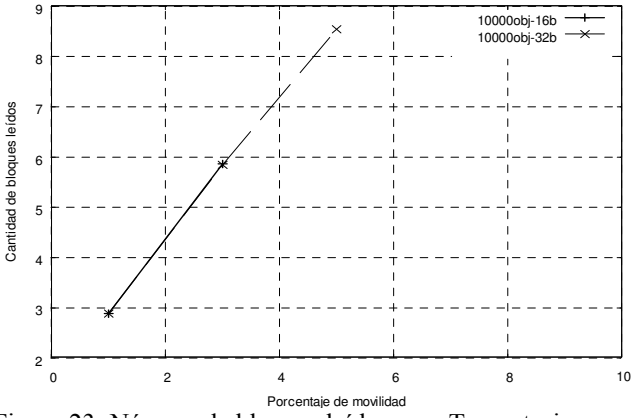


Figura 23: Número de bloques leídos para Trayectoria en un área de 10% para intervalos de 10 unidades para 10000 objetos

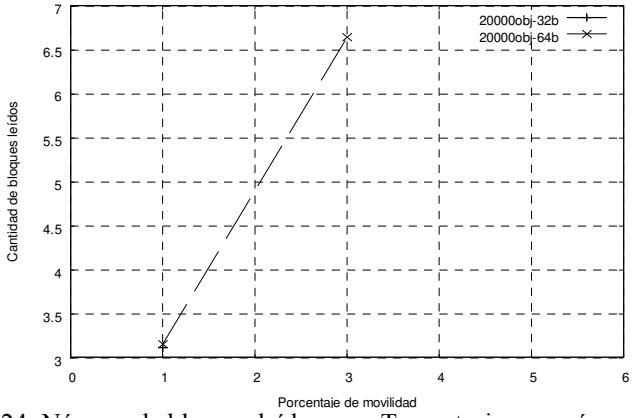


Figura 24: Número de bloques leídos para Trayectoria en un área de 10% para intervalos de 10 unidades para 20000 objetos

REFERENCIAS BIBLIOGRÁFICAS

[1] Dorzán M., Gagliardi E., Gómez Barroso J., y Gutiérrez Retamal G. D*R-Tree: un método eficiente para responder consultas espacio-temporales. CACIC. (2006)

[2] Gutiérrez G. A Spatiotemporal Access Method based on Snapshots and Events. ACM GIS'05, Bremen, Germany. (2005)

[3] Guttman, A. R-Trees: A dynamic index structure for spatial searching. In ACM SIGMOD Conf. on Management of Data, pages 47-57, Boston, ACM. (1984)

[4] Güting, R.H., An introduction to Spatial Database System. VLDB Journal (1994)

[5] Mokbel M., Ghanem T., Aref W. Spatio-temporal Access Methods, IEEE Data Engineering Bulletin 26, pp. 40-49. (2003)

[6] Pfoser D., Jensen C., and Theodoridis Y. Novel Approaches in Query Processing for Moving Object Trajectories. In Proc. of the Intl. Conf. on Very Large Data Bases, VLDB, pages 395–406, (2000)

[7] Porkaew K., Lazaridis I., and Mehrotra S.. Querying Mobile Objects in Spatio-Temporal Databases. In Proc. of the Intl. Symp. on Advances in Spatial and Temporal Databases, SSTD, (2001)

[8] Samet H. Foundations of Multidimensional and Metric Data Structures. ISBN-10: 0123694469. Morgan Kaufmann (2006)

[9] Theodoridis Y., Vazirgiannis M., and Sellis T. Spatio-Temporal Indexing for Large Multimedia Applications. In Proc. of the IEEE Conference on Multimedia Computing and Systems, ICMCS, (1996)

[10] Theodoridis Y., Silva J., and Nascimento M..On the generation of spatiotemporal datasets. (SSD '99), 1999.

Lenguajes de Transformación de Modelos. Un análisis comparativo.

Natalia Correa
LIFIA, Facultad de Informática, Universidad de La Plata
La Plata, Argentina
natalia.correa@lifia.info.unlp.edu.ar

and

Roxana Giandini
LIFIA, Facultad de Informática, Universidad de La Plata
La Plata, Argentina
roxana.giandini@lifia.info.unlp.edu.ar

Abstract

The use of models and transformations among them are the primary concepts of OMG's proposal: MDA (Model Driven Architecture). Specifying transformation has become one of the most important tasks for the proposal, as evidenced by the development of several transformations languages. Studying these languages represents a useful tool for the computing science community whose desire is getting involved with MDA. The goal of this paper is to present a taxonomic framework for analyzing model transformation languages and then to make a comparison among some proposed languages (its features) by using such taxonomy. It is expected that this study reports on benefits to the state of the art of model transformation languages.

Keywords: Software Engineering, Model Driven Architecture, Model Transformation.

Resumen

La utilización de modelos y las transformaciones entre los mismos son conceptos primarios en la propuesta MDA (Arquitectura Dirigida por Modelos) de la OMG. Especificar transformaciones se ha vuelto una de las tareas más importantes, lo cual se ve reflejado en la creación de muchos lenguajes de transformación de modelos. Estudiar y conocer estos lenguajes resulta sumamente útil para toda la comunidad informática que desee comenzar a involucrarse y conocer más sobre esta propuesta. El objetivo de nuestro artículo es presentar un esquema taxonómico para analizar lenguajes de transformación de modelos y evaluar comparativamente algunos de ellos (sus características) en base a esta taxonomía. Esperamos que el análisis realizado constituya un aporte al estado del arte de los lenguajes actualmente definidos.

Palabras claves: Ingeniería de Software, Arquitectura Dirigida por Modelos, Transformación de Modelos.

1 INTRODUCTION

La propuesta MDA (Model Driven Architecture- Arquitectura Dirigida por Modelos)[1] de la OMG (Object Management Group)[2] presenta un proceso de desarrollo de software concebido para dar

soporte al desarrollo de sistemas, donde los conceptos más importantes son los modelos y las transformaciones entre ellos que generan a su vez, otros modelos. Estos se convierten, entonces, en los guías del desarrollo de software. En primer lugar, y según es propuesto por MDA, se definen uno o más modelos PIM (Platform-Independent Model) en algún lenguaje específico y que son independientes de cualquier plataforma de desarrollo. Estos PIMs se traducen en uno o más modelos PSM (Platform-Specific Model) que son específicos de la plataforma donde se ejecuten.
Esta "traducción" entre PIMs y PSMs se conoce como "transformación de modelos".
Describir transformaciones de modelos requiere de lenguajes específicos para la definición de las mismas. Actualmente, existen varias propuestas de lenguajes, muchas de ellas basadas en el estándar de la OMG, QVT (Query/View/Transformation) [3].
Entre los lenguajes definidos existen gran variedad de "tipos": icónicos y textuales, declarativos, operacionales y declarativos-operacionales, algunos basados en QVT y no otros no, compatibles con MOF[4], aquellos que soportan OCL[5], los que proveen traceability, los que proveen composición de transformaciones y hasta aquellos a los que se les ha implementado una herramienta CASE o un plugin para Eclipse.
Estudiar y conocer los lenguajes de transformación que sirven de soporte a la propuesta MDA es útil a toda la comunidad de informáticos. El objetivo de este artículo es, en base al estudio de los lenguajes de transformación, proponer una extensión a una clasificación de lenguajes existente y analizar algunos de los lenguajes que más crecimiento han tenido y que son los más utilizados y nombrados en la comunidad MDA, tomando a esta clasificación como referencia.
La organización del artículo es la siguiente: en la sección 2 presentamos algunos de los lenguajes existentes e introducimos una clasificación para el estudio de los lenguajes de transformación. En la sección 3, realizamos una comparación de lenguajes completando la taxonomía definida en la sección 2 con una selección de lenguajes de transformación de modelos definidos; para finalizar, en la sección 4 presentamos las conclusiones y líneas de trabajo futuro.

2 ESQUEMA DE CLASIFICACIÓN PARA LOS LENGUAJES DE TRANSFORMACIÓN

Desde la aparición de la metodología MDA, mucho se ha propuesto y definido en cuanto a lenguajes y herramientas que sirven de soporte y automatizan sus diferentes aspectos. Uno de estos aspectos donde se ha puesto más énfasis es en la definición de lenguajes que permiten traducir un modelo en otro, pasando desde PIMs a PSMs según indica MDA.

Antes de continuar, enunciamos algunas definiciones de conceptos de lenguajes de transformación. En la jerga MDA, una *transformación* es la generación automática de un modelo de salida o target a partir de un modelo de entrada o source y de acuerdo a la *descripción de una transformación*. Esta descripción se compone de una o más *reglas de transformación* que describen cómo un modelo source puede transformarse en un modelo target (cada uno en sus respectivos lenguajes). Finalmente, una *regla de transformación* describe cómo un elemento del source puede ser transformado en uno o más elementos del target.

2.1 Lenguajes de Transformación de Modelos. Propuestas Existentes

Nuestro trabajo comenzó con la investigación de los lenguajes existentes, muchos de los cuales son presentados a continuación. Como se verá, las propuestas actuales son muchas y muy diversas. Esto hace que no resulte una tarea sencilla la toma de decisión sobre qué lenguaje utilizar. La lista de

lenguajes incluye una breve referencia de cada uno de ellos. Para su posterior análisis, sólo fueron seleccionados algunos de los mencionados.

Tabla 1.Lenguajes de transformación de modelos. Una lista preliminar

Lenguaje	Características
ATL (Atlas Transformation Language) [6]	Lenguaje de transformación de modelos y herramienta en Eclipse desarrollada por el Atlas group (INRIA)
BOTL(Basic Object-Oriented Transformation Language) [7]	Propuesta gráfica de la Universidad Técnica de München
GreAT (Graph Rewriting and Transformations) [8]	Basado en la transformación de grafos. Es la propuesta de una organización independiente: ESCHER Research Institute (The Embedded Systems Consortium for Hybrid and Embedded Research)
JMI (Java Metadata Interface) [9]	Propuesta de Sun basado en MOF que permite manipulación de archivos XMI.
Kent o KMTL(Kent Model Transformation Language)[10]	Propuesta realizada por la Universidad de Kent
MTRANS [11]	Proyecto de la Universidad de Nantes. Es un framework que permite expresar transformaciones de modelos.
Mod-Transf [12]	Lenguaje de transformación de modelos y herramienta en Eclipse desarrollada por el Dart team (INRIA)
MOFScript [13]	Lenguaje de transformación modelo a texto (cuya propuesta pertenece a la OMG) y herramienta como plugin para Eclipse
MOLA (MOdel transformation LAnguage) [14]	Lenguaje gráfico para describir transformaciones propuesto por la Universidad de Letonia.
MT model transformation language [15]	Basado en QVT y desarrollado como DSL (Domain Specific Language) por L. Tratt del King's College de Londres.
MTL (Model Transformation Language) [16]	Lenguaje de transformación de modelos y herramienta en Eclipse desarrollada por el Triskell team (INRIA)
QVT (Query/View/ Transformation) [3]	Especificación estándar de OMG. Está basado en MOF (Meta Object Facility) para lenguajes de transformación en MDA.
Stratego [17]	Lenguaje de descripción de transformaciones de programas. La herramienta desarrollada como soporte del lenguaje es Stratego/XT
Tefkat [18]	Lenguaje declarativo basado en MOF y QVT. Es el aporte de la Universidad de Queensland.
UMLX [19]	Lenguaje gráfico que extiende a UML y también a QVT.
xUML (eXecutable UML) [20]	Propuesta basada en UML para la construcción de modelos de dominio ejecutables y sus transformaciones.

2.2 Clasificación Para La Evaluación de Lenguajes de Transformación

En la sección anterior se ha podido ver que las propuestas de lenguajes de transformación de modelos son muchas y muy diversas. Es claro que a la hora de describir una transformación

aparece un abanico de opciones de las cuales elegir. Y surge, como trabajo adicional, la elección de un lenguaje. Resulta necesario entonces conocer las características de estos lenguajes. Esta necesidad fue vislumbrada por Czarnecki y Helsen [21], quienes propusieron una clasificación para lenguajes de transformación de modelos.
En nuestra propuesta, tomamos como base el trabajo realizado por ellos, reordenando algunas características y adicionando otras.
En la figura 1 puede observarse en color rosa la propuesta original y en color celeste la extensión sugerida en este artículo.

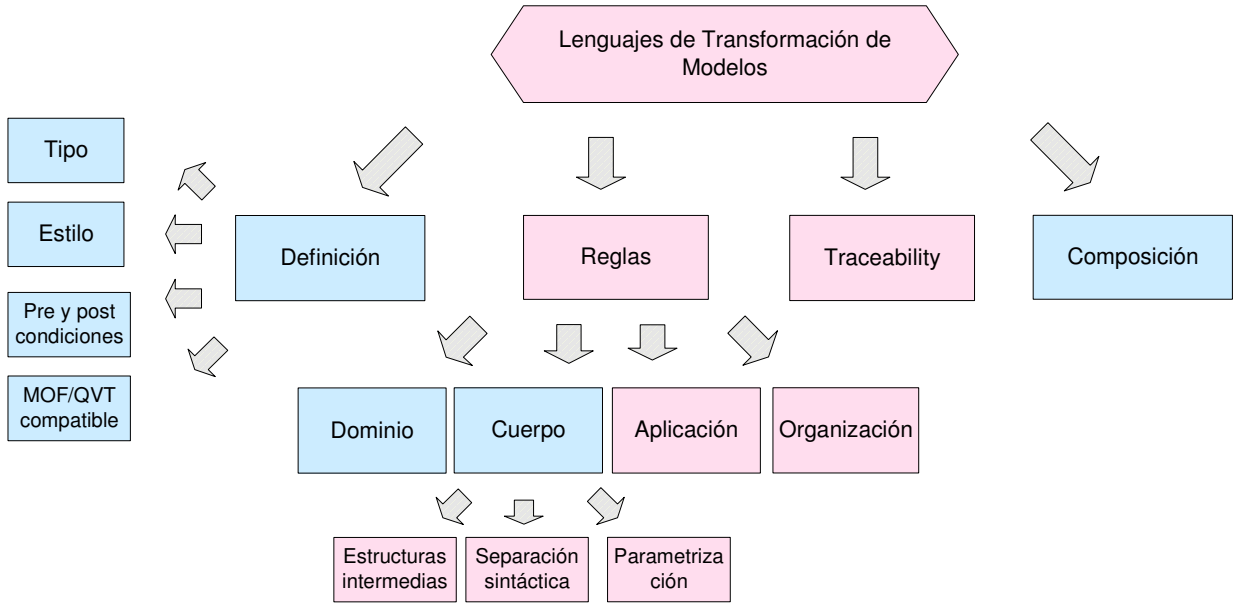


Figura 1. Clasificación propuesta

A continuación se presenta cada una de los aspectos representados en la figura 1, indicando a qué hace referencia cada uno.

Definición

La definición de una transformación determina características generales de la transformación. Se enuncian a continuación:

- **Tipo**: indica si la sintaxis del lenguaje es icónica (gráfica) o textual y si es modelo a modelo (M2M) o modelo a texto (M2Text).
- **Estilo:** un lenguaje puede ser declarativo, operacional o ambos
- **Pre y post condiciones:** si es posible especificarlos en la transformación y de qué forma puede hacerse (coloquial, con OCL, otro lenguaje). Lo deseable es que se puedan especificar y en lo posible, en OCL ya que es un lenguaje formal y estándar.
- **MOF/QVT compatible:** indica si el lenguaje se basa en los estándares definidos por OMG: si los metamodelos (lenguajes de dominio y codominio) que participan en la transformación son instancias de MOF y si el lenguaje de transformación se basa en QVT. Lo esperado es que los lenguajes sean compatibles con ambas propuestas.

Reglas de Transformación

Las reglas de transformación son unidades más pequeñas que componen una transformación. Como fue mencionado anteriormente, estas describen cómo un elemento del modelo de entrada puede ser transformado en uno o más elementos del modelo de salida.
Los aspectos a analizar son:

- **Dominio**: define uno o más modelos "de entrada" u origen o *source* y uno o más modelos "de salida" o destino o *target* sobre los cuales operarán las reglas de transformación
  - **Lenguajes del dominio**: cada dominio tiene un lenguaje asociado. Se espera que los lenguajes sean instancias de MOF.
  - **Dirección**: indica si los metamodelos source y target son *in*, *out*, o *in/out* (en el sentido en que se usan los parámetros en programación). Y si es unidireccional o bidireccional el sentido de la transformación
  - **Relación entre origen y destino de la transformación**: Si el source y el destino tienen el mismo o diferente modelo.

- **Cuerpo de la transformación**: en cuanto al cuerpo de la transformación, podemos decir que consta de:
  - Declaración de (meta)variables y sus tipos
  - Patrones de transformaciones
  - **Separación sintáctica**: algunos lenguajes separan claramente las partes de una regla de transformación que operan sobre el modelo source (LHS) de las partes que operan sobre el modelo target (RHS). La separación sintáctica hace a la legibilidad del lenguaje.
  - **Estructuras intermedias:** si el lenguaje recurre a alguna estructura adicional para describir la transformación y que no es parte del modelo a transformar.
  - **Parametrización**: el tipo más simple de parametrización es el uso de los parámetros del control que permiten el paso de valores.

- **Aplicación de las reglas**: como hemos mencionado, una regla se aplica a algún elemento del modelo de entrada. Como puede haber más de una regla que "machee" con un elemento particular, se debe definir alguna estrategia para determinar el orden de aplicación de las reglas.
  - **Orden:** la aplicación de las reglas puede ser determinístico o no determinístico.
  - **Aplicación condicional**: en algunas reglas de transformación se puede tener aplicación condicional de las mismas. En estos casos, existe una condición que debe ser verdadera para que la regla se ejecute.
  - **Iteración de reglas**: si el lenguaje provee estructuras de iteración, o mecanismos de recursión.

- **Organización de las reglas**: se refiere a la composición y estructuración de múltiples reglas. Se puede dividir en varios aspectos:
  - **Modularización:** en el caso en que se provea modularización de las reglas (agrupar un conjunto de reglas en un módulo, que puede ser llamado desde otros módulos).
  - **Mecanismos de reuso:** la definición de regla/s en base a otra/s.

Traceability

Hace referencia a aquellos mecanismos provistos por los lenguajes que permiten guardar ciertos aspectos de la ejecución de la transformación. Se pueden crear y mantener "conexiones" o *links* entre elementos del dominio y del codominio que mapean los dominios source y target, cada vez

que una regla de transformación es ejecutada.

Composición

La composición indica cómo pueden relacionarse un número de transformaciones para obtener una nueva. Pueden encadenarse dos o más transformaciones consecutivamente; pueden componerse dos o más transformaciones existentes en una nueva transformación con sus nuevas relaciones o pueden combinarse dos transformaciones de forma tal que se obtienen codominios más amplios.

3 COMPARACIÓN DE LOS LENGUAJES DE TRANFORMACIÓN

En base a los criterios presentados en la sección 2.2, analizamos algunos de los lenguajes presentados en la sección 2.1. Para tal análisis, se definieron dos (2) metamodelos y transformaciones entre los mismos (compuestas por varias reglas) y se utilizó cada uno de los lenguajes seleccionados para describir la transformación. Los ejemplos no son incluidos por razones de espacio.

La elección de estos lenguajes se hizo en base al crecimiento, aceptación y utilización que han tenido en la comunidad MDA. Los seleccionados para el análisis presentado fueron los siguientes: **ATL**, **MOFScript**, **MOLA**, **Tefkat** y **ULMX**. Cabe aclarar que todos los lenguajes elegidos cuentan con la implementación de una herramienta que da soporte a la definición del lenguaje. Excepto MOLA, los desarrollos de tales herramientas se han hecho bajo la plataforma y como plugins para Eclipse. Escapa a este trabajo, el análisis, descripción y clasificación de dichas herramientas.

3.1 Análisis de los Lenguajes Seleccionados

ATL (Atlas Transformation Language)

ATL es un lenguaje de transformación de modelos híbrido que permite, en su definición de transformaciones, especificar construcciones declarativas y operativas. La propuesta es del ATLAS Group del INRIA & LINA, de la Universidad de Nantes y fue desarrollado como parte de la plataforma AMMA (ATLAS Model Management Architecture).
ATL es un lenguaje modelo a modelo híbrido o mixto, en el sentido que permite construcciones tanto declarativas como imperativas.
Otros detalles de la Definición del lenguaje:

- Es compatible con los estándares de la OMG: es posible describir transformaciones modelo a modelo (y ambos deben ser instancias de MOF). Además, el lenguaje se basa en QVT.
- Permite definir pre y post condiciones en un lenguaje ya conocido: OCL

En cuanto a las reglas de transformación, ATL define un dominio (metamodelo) para el source y otro dominio para el target, siendo ambos instancias de MOF y con direcciones in y out respectivamente. Source y target pueden tener iguales o diferentes dominios (aunque sean iguales, ambos deben ser claramente identificados). Si bien las transformaciones son unidirecciones, ATL permite la definición de transformaciones bidireccionales como la implementación de dos transformaciones, una para cada dirección.

Como características particulares del lenguaje, podemos mencionar las estructuras que define este lenguaje. En primer lugar, la definición de transformaciones forman módulos (modules) que contienen las declaraciones iniciales y un número de *helpers* y reglas de transformación. Los helpers, son una estructura intermedia dentro de las transformaciones que facilitan la navegación, la modularización y el reuso. Permiten definir operaciones y tuplas OCL Existe también una construcción llamada *called rule,* y que representa a un procedure. Estas pueden contener argumentos y pueden ser invocadas por su nombre. Resulta sumamente expresivo y de fácil escritura para quienes conocen OCL (no es necesario aprender un nuevo lenguaje).

La aplicación de las reglas se realiza de forma no determinística, por "macheo" de reglas y no se ha provisto ninguna construcción o claúsula que permita aplicar en forma condicional las reglas. Cabe mencionar que la invocación de called rules es determinística. Esta invocación, junto con la utilización de parámetros permiten soportar recursión.

ATL provee modularización por sus procedimientos o called rules, además de que sus módulos pueden incluir a otros; y mecanismos de reuso ya que las reglas se pueden heredar.

En cuanto a la trazabilidad, este lenguaje crea un *traceability link* con la ejecución de cada regla que se guarda en el motor de la transformación. Este link relaciona a tres (3) elementos: la regla, el "macheo" (los elementos del source) y los elementos creados en el target.

ATL permite componer transformaciones mediante la definición de reglas.

Para finalizar, vale la pena mencionar que ATL se ha convertido en un lenguaje tan utilizado que ya se ha definido un repositorio de transformaciones entre diversos lenguajes (http://www.eclipse.org/m2m/atl/atlTransformations/) .Ya existen más de 60 trasformaciones documentadas y con los archivos fuente disponibles.

MOFScript

MOFScript es un lenguaje de transformación de modelo a texto presentado por la OMG. Este lenguaje presta particular atención a la manipulación de strings y de texto y al control e impresión de salida de archivos.
No sólo es un estándar, sino que además se basa en otros estándares de la OMG: es QVT compatible y MOF compatible con el modelo de entrada (el target siempre es texto).
Para las reglas de transformación, MOFScript define un metamodelo de entrada para el source y sobre el cual operarán las reglas. El target es generalmente un archivo de texto (o salida de texto por pantalla).
Las transformaciones son siempre unidireccionales y no es posible definir pre y post condiciones para ellas. La separación sintáctica resulta clara por la misma definición de reglas, lo que hace a la legibilidad del lenguaje. No provee estructuras intermedias, pero sí parametrización necesaria para la invocación de reglas.

En cuanto a la aplicación de reglas, podemos decir que se aplican en forma determinística y en orden secuencial. Se provee aplicación condicional e iteración de reglas. Las condiciones de aplicación se expresan en cláusulas *when*, como definición de guardas. La iteración se realiza mediante los iteradores *for each* y *while*.

MOFScript no organiza las reglas en módulos propiamente dichos. Ahora bien, en la definición de una regla se puede invocar a otras reglas, utilizando incluso parámetros, con lo cual se asemeja

bastante a un módulo. Y es posible definir jerarquías de transformaciones.

Finalmente, para *traceability*, MOFScript define (pero no implementa aún) un conjunto de conceptos para relacionar elementos del source con sus ubicaciones en los archivos de texto generados en el target.
En cuanto a la composición de transformaciones, no es algo que se haya pensado en este lenguaje aún.

MOLA (MOdel transformation LAnguage)

MOLA es un lenguaje gráfico de transformación de modelos, propuesto por la Universidad de Letonia. La intensión de este lenguaje es combinar la programación estructurada tradicional con reglas basadas en patrones de transformación.

La definición del lenguaje no se basó en QVT y los metamodelos que participan de la transformación pueden o no ser instancias de MOF; lo único restrictivo es que ambos deben ser definidos como dos modelos diferentes (source y target) aunque sean el mismo. El source es el modelo de entrada in/out (sus elementos pueden ser modificados) y el target de salida (sólo out). Las transformaciones son unidireccionales.
Si bien no se mencionan pre y post condiciones, el lenguaje provee notas donde pueden escribirse cláusulas OCL para la aplicación condicional de reglas. Estas podrían usarse entonces para pre y post condiciones.

En cuanto al cuerpo de la transformación, se ve que no hay separación sintáctica de los elementos que se transforman y de los que se crean. Tampoco provee estructuras intermedias.
El elemento principal del lenguaje es un concepto gráfico del *loop*, que se utiliza mucho para iterar sobre los elementos de los modelos que participan de la transformación. También para ello, permite definir variables locales a cada regla de transformación.

Sobre la aplicación de reglas, podemos decir que se aplican en forma determinística y en orden secuencial según son definidas. Existe la aplicación condicional mediante la adición de notas con cláusulas OCL. La iteración se realiza mediante dos tipos de loop: un loop de "tipo uno" que se ejecuta una vez para cada instancia válida del source (for each); un loop de "tipo dos" que continúa la ejecución mientras haya al menos una instancia válida en el source (while). El lenguaje utiliza patrones definidos para ser aplicados en las transformaciones.
Si bien gráficamente resulta intuitivo, las iteraciones pueden volverse en poco confusas, sobre todo si hay varias reglas anidadas.

En cuanto a *traceability*, MOLA permite definir "*mapping associations*" para tracear instancias entre modelos. Estas se definen en las reglas como notas anexadas entre los elementos del dominio y codominio que participan de la transformación.

La composición de transformaciones no ha sido tenida en cuenta en la definición del lenguaje.

Tefkat

Tefkat es la definición e implementación de un lenguaje para transformación de los modelos. La propuesta fue realizada por de la Universidad de Queensland, Australia y se encuentra en una etapa de desarrollo bastante avanzada con respecto a otros lenguajes.

El lenguaje transforma modelos (M2M) en forma textual y ha adoptado un paradigma declarativo. Tefkat es totalmente compatible con los estándares de la OMG. Entiende que los modelos source y target definen dominios diferentes para cada uno de ellos, siendo ambos instancias de MOF.
Como desventaja, podemos mencionar que no hay declaración de pre ni de post condiciones.
Introduciéndonos en el cuerpo de la transformación, vemos que es posible declarar variables y metavariables, y que la separación sintáctica es provista por el lenguaje. De hecho, le aporta mucha legibilidad y claridad a la escritura de las reglas. Existen estructuras FORALL (algo de source) MAKE (algo del target) y FROM (algo de source) TO (algo del target) más la identación de las sentencias.
Tefkat también provee patrones (para el source) y templates (para el target) que se utilizan para nombrar restricciones que se pueden utilizar en más de una regla. Estas construcciones pueden a su vez parametrizarse, permitiendo la invocación a patrones y la recursión.

Las reglas se aplican en forma no determinística. Existe la aplicación condicional soportada por la cláusula IF-THEN-ELSE que permite la ejecución de reglas cuando la guarda es evaluada como verdadera. Las reglas pueden iterar y ejecutarse con recursión.

Para *traceability*, Tefkat incluye una cláusula LINKING que representa un *mapping* (asociación) entre los elementos del source y del target que participan de la transformación y que son almacenados una vez realizada la misma.

En este lenguaje es posible componer transformaciones mediante la definición de reglas, tomando transformaciones existentes y creando una nueva que preserva las relaciones.

UMLX

UMLX es un lenguaje gráfico de transformaciones entre modelos (M2M) y que se basó en extensiones mínimas a UML. Es una propuesta de E. Willink, del GMT Consortium. En un último avance, se ha anunciado que la transformación textual que la herramienta UMLX traduce desde el gráfico, puede también editarse. Como se traduce a lenguaje OCL, sería muy conveniente para aquellos que ya conocen este lenguaje y no se cae en la necesidad de conocer otro lenguaje.

La definición de este lenguaje se basó en QVT. Asimismo, los modelos participantes de las transformaciones, deben ser instancias de MOF. Cada uno de estos es tomado como un dominio diferente (uno de entrada o in y otro de salida o out), aunque se trate de los mismos modelos. Para estos no pueden especificarse pre o post condiciones.

En cuanto a las reglas de transformación, UMLX utiliza diferentes íconos gráficos para las transformaciones, para las reglas y para las relaciones de creación, preservación y eliminación de elementos. Cabe destacar que la semántica de los íconos del lenguaje no se ha definido, y aunque es gráficamente intuido, para algunas relaciones, no queda claro cuál es su significado.

Las estructuras que define este lenguaje son bastante simples e intuitivas, pero carecen de una semántica bien definida. No hay módulos, estructuras intermedias ni cláusulas de iteración o condición. Sólo existen variables locales, para indicar que se hace referencia a una instancia en particular. Por ejemplo, si se define una regla para todas las clases, y se define una regla también para cada atributo, al hacer referencia a una instancia de clase particular (cl, por ejemplo), se pueden mencionar a todos los atributos de cl, notándolo con @cl

La aplicación de las reglas se realiza de forma no determinística, por *pattern matching* de reglas y no se ha provisto ninguna construcción o claúsula que permita aplicar en forma condicional las reglas. Tampoco pueden invocarse otras reglas, ni se utilizan parámetros.

Ni la trazabilidad ni la composición han sido agregadas al lenguaje.

En las tablas 2 y 3 se presenta un resumen de las características enunciadas anteriormente y a modo de comparación directa entre los distintos lenguajes seleccionados.

Tabla 2. Comparación de lenguajes.

Característica/ lenguaje			**ATL**	**MOFScript**	**MOLA**
DEFINICIÓN	Tipo		Textual/ M2M	Textual/ M2Text	Gráfica/ M2M
	Estilo		Declarativo y operacional	Declarativo y operacional	Declarativo y operacional
	Pre y post condiciones		Si (OCL)	No	No
	MOF/QVT compatible		MOF: sí QVT: sí	MOF: sí QVT: sí	
REGLAS DE TRANSFORMACIÓN	DOMINIO	Lenguajes de dominio	Sí	Sí	Sí
		Dirección	Uni y bidireccional	Unidireccional	Unidireccional
		Relación entre origen y destino	Mismos o diferentes modelos	Diferentes modelos	Mismos o diferentes modelos
	CUERPO	Declaración de (meta)variables	Sí	Sí	Sí
		Patrones de transformaciones	No	No	Sí
		Separación sintáctica	Sí	Sí	No
		Estructuras intermedias	Sí	No	No
		Parametrización	Sí	Sí	Sí
	APLICACIÓN	Orden	No determinístico	Determinístico	Determinístico
		Aplicación condicional	No	Sí	Sí
		Iteración de reglas	Sí	Sí	Sí
	ORGANIZAC	Modularización	Sí	No	Sí
		Mecanismos de reuso	Sí	Sí	No
TRACEABILITY			Sí	Sí	Sí
COMPOSICIÓN			Sí	No	No

Tabla 3. Comparación de lenguajes (continuación).

Característica/ lenguaje			Tefkat	UMLX
DEFINICIÓN	Tipo		Textual/ M2M	Gráfico/ M2M
	Estilo		Declarativo	Declarativo y operacional
	Pre y post condiciones		No	No
	MOF/QVT compatible		MOF: sí QVT: sí	MOF: sí QVT: sí
REGLAS DE TRANSFORMACIÓN	DOMINIO	Lenguajes de dominio	Sí	Sí
		Dirección	Unidireccional	Unidireccional
		Relación entre origen y destino	Mismos o diferentes modelos	Mismos o diferentes modelos
	CUERPO	Declaración de (meta)variables	Sí	Sí
		Patrones de transformaciones	Sí	No
		Separación sintáctica	Sí	No
		Estructuras intermedias	No	No
		Parametrización	Sí	No
	APLICACIÓN	Orden	No determinístico	No determinístico
		Aplicación condicional	Sí	No
		Iteración de reglas	Sí	No
	ORGANIZAC	Modularización	Sí	No
		Mecanismos de reuso	Sí	No
TRACEABILITY			Sí	No
COMPOSICIÓN			Sí	No

4 CONCLUSIONES Y TRABAJO FUTURO

El desarrollo de software dirigido por modelos en una metodología que ha cobrado impulso desde su aparición en 2003. Tanto es así, que muchos trabajos se han propuesto y definido en cuanto a lenguajes, frameworks y herramientas que sirven de soporte y automatizan diferentes aspectos de MDA. Al adherirse a esta propuesta, es necesario elegir algunos de los lenguajes, lo cual no resulta una tarea sencilla. Nuestro trabajo ha presentado una extensión a un esquema de clasificación existente que nos permite evaluar características de los lenguajes de transformación de modelos. A su vez, realizamos una investigación sobre las diversas propuestas y seleccionamos algunas para evaluarlas en base a la clasificación antes mencionada.

De este análisis se desprende que tanto ATL como Tefkat son lenguajes muy completos y en avanzado desarrollo. Ambos adhieren a los estándares de la OMG, utilizan un lenguaje formal como es OCL y proveen mecanismos para traceability y composición de las reglas de transformación. ATL hace uso además de estructuras intermedias que mejoran la legibilidad y el reuso. Dentro de los lenguajes gráficos, MOLA es una buena propuesta: con buena legibilidad, intuitiva, fácil de

comprender y de utilizar.
Como líneas de trabajo futuro, podemos mencionar las siguientes: la mejora de ciertos aspectos a la propuesta de lenguaje mínimo para transformaciones de modelo que el grupo de investigación está desarrollando, en base a los lenguajes analizados; y la incorporación a esta misma propuesta de mecanismos que soporten traceability. Ambas líneas, se integrarían a la herramienta CASE [22] que nuestro grupo de investigación está desarrollando.

REFERENCIAS

[1] MDA Guide, v1. 0. 1, omg/03-06-01, June 2003. http://www. omg. org.
[2] OMG (Object Management Group) http://www. omg. org
[3] MOF 2. 0 Query/View/Transformations (QVT) - OMG Adopted Specification. March 2005. http://www. omg. org.
[4] Meta Object Facility (MOF) 2. 0. OMG Adopted Specification. 2003. http://www. omg. org.
[5] OMG. The Object Constraint Language Specification – Version 2. 0, for UML 2. 0, revised by the OMG, http://www.omg.org, April 2004.
[6] Jouault F. , Kurtev I. Transforming Models with ATL Workshop in Model Transformation in Practice at the MoDELS 2005 Conference. Montego Bay, Jamaica, Oct 3, 2005
[7] Marschall, F., Braun, P.: BOTL - The Bidirectional Object Oriented Transformation Language. Instituto de Informática, Universidad Técnica de Munich. Munich (2003)
[8] Agrawal, A., Kalmar, Z., Karsai, G., Shi, F., Vizhanyo, A.: GReAT User Manual.Nashville: Institute for Software-Integrated Systems, Vanderbilt University (2003)
[9] Sun Developer Network: Java Metadata Interface (JMI). SUN (2002)
http://java.sun.com/products/jmi/
[10] Akehurst, D.H., Howells, W.G., McDonald-Maier K.D.: Kent Model Transformation Language. En: MoDELS 2005 Conference. Montego Bay, Jamaica (2005)
[11] M. Peltier, J. Bézivin, and G. Guillaume. MTRANS: A general framework based on XSLT for model transformations. In WTUML'01, Proceedings of the Workshop on Transformations in UML, Genova, Italy, April 2001
[12] Model transformation- Inria. Mod-Transf ('04). http://modelware.inria.fr/rubrique15.htlm
[13] Eclipse org & Modelware. MOFScript (2005). http://www.eclipse.org/gmt/mofscript/
[14] Kalnins A., Barzdins J., Celms E. Model Transformation Language MOLA. Proceedings of MDAFA 2004, University of Linkoping, Sweden, 2004, pp.14-28.
[15] Tratt, L. The MT model transformation language. In MT 2006, Proceedings of the 2006 ACM symposium on Applied computing, pages 1296 - 1303
[16] Akehurst D , Howells W. , McDonald-Maier K. Model Transformation Language. Workshop in Model Transformation in Practice - MoDELS 2005 Conference, Jamaica, Oct 3, 2005
[17] Program-Transformation.Org. Stratego: Strategies for Program Transformation. Program-Transformation (2004). http://www.strategolanguage.org/Stratego/WebHome
[18] Lawley M. , Steel J. Practical Declarative Model Transformation with TefKat. Workshop in Model Transformation in Practice - MoDELS 2005 Conference. Jamaica, Oct 3, 2005
[19] Willink, E. UMLX - A graphical transformation language for MDA. En: OOPSLA 2003 Conference. Anaheim, California (2003)
[20] eXecutable UML.OMG proposal (2005). http://www.omg.org/cgi-bin/doc?ad/2005-4-2
[21] Czarnecki, Helsen. Feature-based survey of model transformation approaches.IBM System Journal,V45,N3, 2006
[22] Pons C., Giandini R., Pérez G., et al. Precise Assistant for the Modeling Process in an Environment with Refinement Orientation. "UML Modeling Languages and Applications: UML 2004 Satellite Activities, Revised Selected Papers". LNCS #3297. Springer, 2004.

"Modelo para el entrenamiento de usuarios con déficit auditivo a través integración de sonidos con el contexto

Jorge Ierache [1,3], Luís Campos[2,3], Marcela Bruno[1,3] ,Hugo Padovani [3]
[1]Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica.
[2]Instituto de Tecnología de la Información y Comunicación para el Desarrollo Social
[3]Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales.
Universidad de Morón {jierache,mabruno,lcampos,hpadovani}@unimoron.edu.ar
Tel: +5411 56272000 (189/746)

Abstract

A preliminary model for the training of users with severe auditory deficit appears in this work. The model contemplates to the edition of "Contexts of action" for example situations as the visit to a thematic park, zoological, etc; those that are made up of different images and sounds. The images consider associates with dynamic scenes that will contain sounds that represent them. Of this form it is tried that the disabled one can perceive from the simulator the vibrations corresponding to these sounds through a device of stimulation by vibrations the one that presents alternative means of communication for the people with severe auditory deficit, this device and its interaction was developed by J Escudero, L Campos 2004. In relation to mobility the proposed model contemplates the interoperability of the disabled one with contexts of real sounds through the connection of the device of stimulation by movable device vibrations that the user can take with himself, like for example cellular or PDA. The real context becomes rich when incorporating a reader of RFID to the movable devices that allow the user to unload information and sounds of the context to their cellular or PDA, from a physical URL.

Key words: Engineering Centered in Contexts, Auditory Disabled, Tactile-vibration Stimulation, Model of trainer

Resumen

Se presenta en este trabajo un modelo preliminar para el entrenamiento de usuarios *con déficit auditivo severo*. El modelo contempla la edición de "Contextos de acción" por ejemplo situaciones como la visita a un parque temático, zoológico, etc; los que se encuentran compuestos por diferentes imágenes y sonidos. Las imágenes se consideran asociadas con escenas dinámicas que contendrán sonidos que las representen. De esta forma se pretende que el discapacitado pueda percibir desde el simulador las vibraciones correspondientes a estos sonidos a través de un dispositivo de estimulación por vibraciones el que presenta un medio alternativo de comunicación para las personas *con déficit auditivo severo*, este dispositivo y su interacción fue desarrollado por J Escudero, L Campos 2004. En relación a la movilidad el modelo propuesto contempla la interoperabilidad del discapacitado con contextos de sonidos reales a través de la conexión del dispositivo de estimulación por vibraciones a dispositivos móviles que el usuario puede llevar consigo, como por ejemplo un celular o una PDA. El contexto real se enriquece al incorporar un lector de RFID a los dispositivos móviles, que permitan al usuario descargar información y sonidos del contexto a su celular o PDA, desde una URL física.

Palabras Claves: Ingeniería centrada en contextos, Discapacidad auditiva, Estimulación vibrotactil, Modelo de entrenador

1. INTRODUCCIÓN

Se realizan diversos esfuerzos en la actualidad para ayudar a las personas con discapacidades ,el trabajo realizado por J Boyd Graber et al para personas que sufren de Aphasia [5], (desorden que implica dificultades para hablar leer escribir y entender el lenguaje) presenta el diseño y una evaluación preliminar de una aplicación de desktop-handheld para ayudar a individuos con Aphasia, desarrollando una comunicación a través de imágenes y sonidos sobre una computadora desktop y dispositivos móviles para soportar la comunicación fuera de casa. En esta línea se destaca el trabajo desarrollado por J Escudero, L Campos 2004 para discapacitados con déficit auditivo severo, donde los resultados que se han obtenido con el uso del dispositivo de estimulación vibrotáctil [1], facilitaron a los usuarios aprenden con rapidez a identificar y reconocer palabras, frases y también diversos sonidos del ambiente que los rodea, así como también la lectura de los labios.

El presente trabajo contribuye con el trabajo de J Escudero y L Campos, aportando una propuesta de modelo para incorporar sonidos del contexto, a fin de facilitar el entrenamiento de discapacitados, el modelo pretende generar un contexto para los diversos sonidos que se pueden producir en un determinado ambiente. Por ejemplo, en la calle, se escucharán sonidos de motores de autos, de motos, bocinas, personas caminando, etc. En un zoológico en cambio, escucharemos sonidos de diversos animales, etc.

La aplicación propuesta es centrada en Contexto, de acuerdo a lo planteado por Dey en el trabajo de A Fortier, G Grossi, S Gordillo [3] contexto es alguna información que puede ser usada para caracterizar una situación de una entidad, una entidad es una persona, un objeto, un lugar que es considerado relevante para la interacción entre el usuario y una aplicación, incluyendo al usuario y sus aplicaciones.

La visión de contexto fenomenológica que presenta Dourish de acuerdo a lo planteado por Dey en el trabajo de A Fortier, G Grossi, S Gordillo [3] el foco es el ser humano, que percibe el contexto en nuestro caso con discapacidad auditiva, se encuentra en un constante proceso de redescubrimiento, en nuestro caso el aprendizaje de nuevos sonidos del contexto, el contexto finalmente para esta visión es subjetivo y se acerca a definiciones abstractas; para esta visión el contexto no es información, habla de una relación contextualizada entre objetos o actividades, el contexto varia no puede modelarse de antemano, el contexto es una propiedad ocasionada, depende de las situaciones, el contexto surge de la actividad, no hay distinción entre actividad (comportamiento) y contexto (datos), para esta visión resulta necesario determinar un modelo que describa el contexto y que maneje los cambios de este, en nuestro caso los contexto que representan el ambiente y sus sonidos.

2. JUSTIFICACIÓN:

El uso del dispositivo de estimulación por vibraciones, abre un nuevo mundo de posibilidades de comunicación y de percepciones a los usuarios hipoacúsicos. Sin embargo existe la dificultad, que para entrenarse en la percepción de sonidos de diferentes contextos, el usuario debe trasladarse al lugar físico donde se producen esos sonidos para poder percibirlos. Esto sin duda es una desventaja, ya que según el contexto de que se trate, puede ser poco accesible para el usuario e implica un esfuerzo el tener que movilizarse hasta el lugar. Por otra parte esto implica disponer de un tiempo limitado para entrenarse con los sonidos de ese contexto, y de no tener el usuario la posibilidad de "repetir" aquellos sonidos que tuvo más dificultad en reconocer.

Surge así la necesidad de disponer de una interfaz, que facilite el aprendizaje de sonidos de contexto, sin la necesidad de tener que ir hasta un determinado lugar. El usuario podrá recorrer escenarios creados virtualmente, que contendrán imágenes correspondientes al mismo. Estas imágenes, a su vez, están asociadas a un escenario dinámico, representado por imágenes y sonidos. Por otra parte cuando el usuario se encuentre en un escenario real en donde existen objetos físicos, este tendrá la posibilidad de descargar información a través etiquetas RFID [4] que se encuentren en

los objetos del ambiente, las que facilitaran la información a través de URLs físicas. De esta forma, cuando el usuario recorra un determinado lugar, en donde existen objetos con etiquetas RFID, tendrá la posibilidad de descargar la información y sonidos a su celular o PDA. Esta información podrá aportar una explicación sobre el objeto en cuestión, (por ejemplo si está en un zoológico y está frente a el hábitat de una animal, podría descargar toda la información referente a este animal, pero fundamentalmente los sonidos característicos (Por ejemplo en el caso de un paseo por el zoológico, el RFID de la jaula de los leones, podría contener el sonido del rugido del león). Todos estos sonidos, podría percibirlos a través del dispositivo de estimulación por vibraciones.

La interfaz de usuario facilitara el armado de contextos de acción, con imágenes que pueden provenir de Internet, que el usuario mismo puede haber capturado con un celular, etc. Además permitirá recorrido de contextos de acción, en forma general, con la posibilidad de reproducir el nombre de las imágenes que contiene y acceder a los escenarios dinámicos correspondientes. En otro orden serán necesarias funciones de apoyo para almacenar información relevante, relacionada con el nivel entrenamiento y tipo de discapacidad del usuario, como así también posibilidades de bajar nuevos escenarios desde Internet para el entrenamiento.

3. CARACTERÍSTICAS DEL DISPOSITIVO DE ESTIMULACIÓN POR VIBRACIONES:

En esta sección se hará una breve descripción de los aspectos más relevantes en cuanto a principios de funcionamiento y características del dispositivo de estimulación por vibraciones [1,2]. Los sonidos del mundo exterior, se captan por medio de un micrófono, y se transforman en estímulos vibratorios, que el usuario percibe a través del dispositivo, el proceso de traducción de los sonidos a vibraciones en forma general se distinguen tres etapas:

- Generación de la Información: El evento disparador es la recepción de un sonido exterior. (Señal analógica).
- Procesamiento: La señal, sonora recibida, se pasa a valores digitales mediante un conversor A/D, para poder procesarla. La salida resultante se vuelve a pasar a una señal analógica mediante un conversor D/A.
- Recepción: La señal analógica resultante es recibida por el discapacitado auditivo mediante la utilización de una interfaz háptica.

El dispositivo de comunicación por vibraciones consiste en un dispositivo de comunicación por vibraciones para personas con discapacidad auditiva, comprende un vibrador y un receptor, ambos conectados entre sí por medio de un circuito que puede comprender un medio informático, una PC, un amplificador de señal o cualquier otro medio que procese señales captadas por el receptor. El vibrador es un elemento transductor electroacústico, como por ejemplo un microparlante, y está en contacto con el cuerpo a través de la yema del dedo. Este vibrador puede estar conectado con el receptor en forma alámbrica o inalámbrica y el receptor comprende un sensor de ondas magnéticas, un micrófono o similar, del micrófono sale una señal de audio modulada en amplitud, la cual es amplificada para excitar el dispositivo vibrador. El vibrador es de tipo portable; está sujeto a un soporte, constituido por una faja anular elástica con un cierre de velcro, con el que puede fijarse al extremo del dedo índice para mantener la superficie vibradora en contacto permanente y seguro con la yema del dedo. El receptor de señal es portable, y es un micrófono que está conectado a una computadora que almacena sonidos en una biblioteca de sonidos. La persona podrá asociar cada vibración que le es transmitida a través del vibrador, a una letra, palabra o sonido y combinación de los mismos, llegando a interpretar claramente lo que le dicen o los sonidos de un entorno habitual, se requiere de un breve período de aprendizaje, como lo demuestran los ensayos. [2] Este dispositivo puede ser aplicado a personas sordas que también son ciegas. La persona sordo ciega escribe en el teclado Braille y la computadora le devuelve a través del vibrador, las vibraciones correspondientes a lo escrito. [2] El dispositivo es aplicable a un teléfono celular o de línea, en

donde el vibrador estaría fijado a una parte del teléfono que sea accesible al usuario. Como se puede advertir, este mecanismo no sólo comprende un dispositivo de intercomunicación, sino que involucra una nueva tecnología y técnicas de comunicación para personas con discapacidad auditiva, así como nuevos sistemas de enseñanza.

3.1. Interfaz Háptica:

La Interfaz seleccionada para el dispositivo debía ser capaz de poner en contacto al discapacitado auditivo con la información procesada mediante la utilización de alguno de sus sentidos. Tradicionalmente, se ha atribuido al sentido de la vista la mayor importancia en la recepción de estímulos e interacción durante el proceso de comunicación, el oído se define en segundo lugar y por último el tacto. Este esquema de funcionamiento responde a la realidad de nuestra vida cotidiana, en una persona normalmente constituida y sin lesiones de ningún tipo, pero dado que el dispositivo de comunicación estaba orientado a personas con una lesión severa en su sentido auditivo, se propuso [1]. un nuevo esquema de funcionamiento priorizando la percepción de la información procesada mediante el sentido del tacto en ausencia del sentido del oído. Considerando el enorme caudal de información que puede ser recibida a través del tacto. Se denomina háptica a toda aquella percepción táctil realizada en forma activa y voluntaria; es decir, se requiere un uso activo de los dedos y manos en el descubrimiento de objetos de nuestro entorno debido a que el canal de comunicación está centrado en las manos, más específicamente las palmas y dedos. Las interfaces hápticas buscan combinar el sentido del tacto con su entorno, su investigación se puede subdividir en dos campos:

- Retroalimentación de la fuerza: Trata con dispositivos que interactúan con músculos y tendones y dan al humano la sensación de que se aplica una fuerza.
- Retroalimentación táctil: Es la que se usó en el desarrollo del dispositivo de estimulación por vibraciones. Trata con dispositivos que interactúan con los nervios terminales en la piel, los cuales indican la presencia del calor, presión y textura.

3.2. Dispositivo de estimulación vibrotáctil:

Se construyó Centro Argentino de Medios Alternativos de Comunicación (CAMAC) [2], este dispositivo se utiliza en el dedo índice adaptado a la forma y tamaño del dedo índice del discapacitado auditivo, la base del dispositivo tiene una cavidad libre, el objetivo de este diseño es contener el dedo y permitir el contacto de la yema con la vibración proveniente de la señal analógica sonora (Figura 1).

Figura 1: Dispositivo de estimulación por vibraciones

3.3. Interfaz de entrenamiento lingüístico:

El trabajo de Escudero J y Campos L presentó una interfaz de entrenamiento lingüístico (Figura 2), para reconocer los estímulos vibratorios correspondientes a palabras y frases, que permite al instructor administrar la información de los discapacitados auditivos con los que trabaja, estos datos son característicos de la voz digital con la que se estuvo trabajando y el historial de secuencias. Además se permite al instructor personalizar las características de la voz digital a trabajar con el discapacitado auditivo considerando el valor asociado a velocidad, pitch y tiempo. En otro orden se permite al instructor ingresar en un campo de edición las palabras u oraciones. Al mismo tiempo del ingreso, cada palabra se va visualizando en campos de lectura independientes. La interfaz cuenta con diez campos de lectura e incorpora la simulación de los labios correspondiente a las sílabas o palabras.

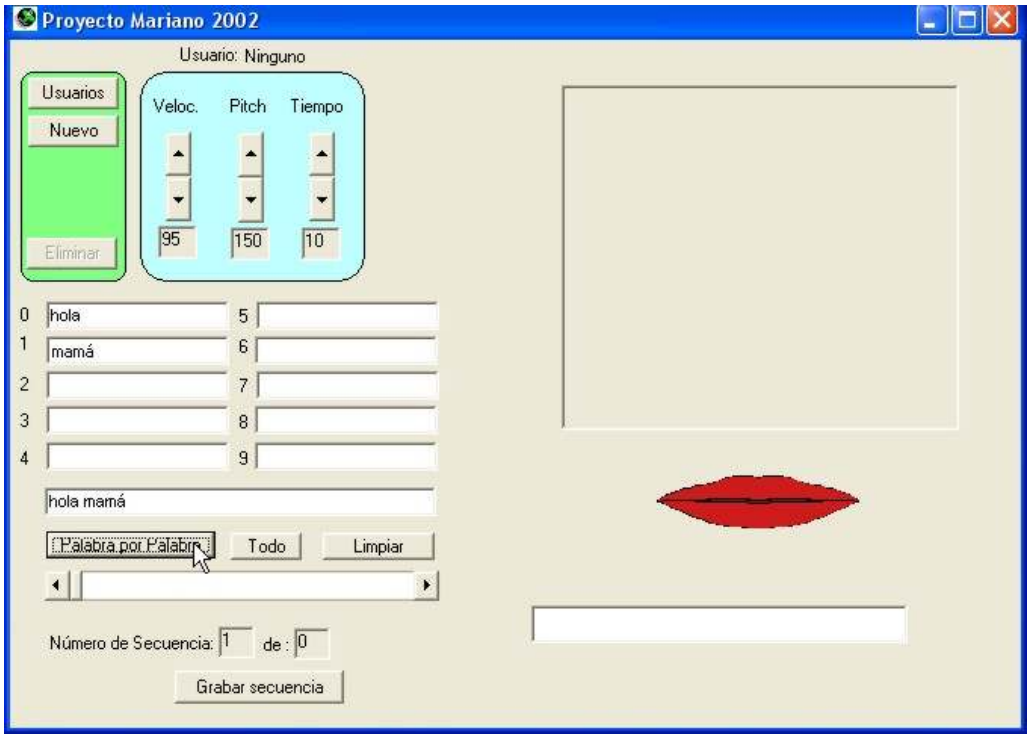


Figura 2: Pantalla principal del software de entrenamiento

4. MODELO PROPUESTO PARA EL ENTRENAMIENTO DE SONIDOS CONTEXTUALES:

4.1. Interfaz de entrenamiento context-awareness

La interfaz propuesta, será una interfaz context-awareness. Para entender la definición de "orientado al contexto" se debe empezar a definir y comprender el concepto de "computación context-awareness". Cuando realizamos una acción, generalmente, es producto de la intención, que tenemos, de alcanzar un objetivo. Estos objetivos pueden ser muy simples para nosotros pero sumamente complejos para una aplicación de software, debido a que esta última, no conoce cuál es nuestro deseo en ese momento, y además, no posee la capacidad de procesar cognitivamente gran

cantidad de aspectos, contextuales (sonidos, luminosidad, espacio, tiempo, etc.) y personales (intensiones, intereses, gustos, preferencias, necesidades, etc.). Estos aspectos los percibimos en forma automática y no nos resultan un obstáculo, pero son un gran desafío para una aplicación. Los investigadores en el área de la ingeniería de software están incorporando estos aspectos en las nuevas aplicaciones logrando que estas, sean sensibles al contexto [10]. Este nuevo paradigma fue denominado computación sensible al contexto o context-awareness.

4.1.1 *Consideraciones acerca de context-awareness.*

A lo largo del tiempo se han formulado diferentes definiciones. La primera definición surge en 1993, (Schilit [11]) define al contexto como "*la localización, identidad de las personas y objetos cercanos y los cambios que se produzcan en estos objetos*", en forma similar, Brown [12] define al contexto como "*localización, identidad de las personas que rodean al usuario, hora del día, estación del año, temperatura, etc*.". Ryan [13] lo presenta como *"la localización del usuario, el entorno físico, la identidad y la hora"*. Siguiendo esta línea, en cuanto a la percepción de los elementos que deben ser considerados parte del contexto, Dey [14], en 1998, brinda una definición más completa, donde ve al contexto como *"el estado emocional del usuario, el foco de atención, la localización y orientación, fecha y hora, y las personas que componen el entorno del usuario"*.
Todas estas definiciones enumeran los "elementos", que para el autor, tendrían que formar parte del contexto, ahora, ¿qué ocurre si se agrega un nuevo elemento?, ¿como saber si forma o no parte del contexto? Un ejemplo de esto podría ser, *"servicios disponibles"*. Al Agregar un nuevo elemento, estas definiciones se vuelven insuficientes y dejan "una zona gris", por tal motivo es necesario encontrar una definición más abstracta y amplia. Brown [15], expreso al contexto como *"los elementos del entorno del usuario de los que la computadora del usuario es consciente"*. Ward [16], lo define sencillamente como "*el estado del entorno de la aplicación"*. Pascoe [17] lo expresa como "*el conjunto de estados físicos y conceptuales de interés para una entidad particular*". La definición de Franklin [18] es sumamente abstracta, caracterizo al contexto como "*la situación del usuario*".
Dey en el artículo "*Towards a Better Understanding of Context and Context-Awareness*" [9], aporta una definición desde el punto de vista del desarrollo de software, caracterizando al contexto como: "*cualquier información que puede ser usada para caracterizar la situación de una entidad. Una entidad puede ser una persona, un lugar o un objeto que es considerado relevante para la interacción entre el usuario y la aplicación; incluyendo al usuario y la aplicación mismos"*.

4.1.2 Entorno de aplicaciónes context-awareness

Todos los seres vivos interaccionan constantemente con su entorno. Con múltiples grados de complejidad, los organismos son capaces de percibir los cambios que se producen en el entorno que los rodea y de reacomodar su comportamiento o su metabolismo para adaptarse a dichos cambios. Un cambio que permita a un ser vivo funcionar eficientemente se llama adaptación. El cambio adaptativo significa una ventaja para vivir en un entorno concreto. B. Schilit y M. Theimer [19] y más tarde (1996), M. Brown [20], todos utilizaron el concepto de "*adaptación*" y coincidiendo en una primera definición que expresaba lo siguiente: "*una aplicación de software es sensible al contexto si puede adaptarse, percibir o responder ante los cambios en el entorno*". Con el tiempo a las aplicaciones context-awareness se les atribuyeron términos como: "*reactivas*" [20], "*obedientes*" [21], "*situadas*" [22], "*sensible al contexto*" [23] y "*dirigidas por el ambiente*" [24]. Kortuem [25], tomo algunos de estos conceptos y definió este tipo de aplicaciones como "*aquellas que pueden variar o adaptar dinámicamente su comportamiento en base al ambiente*". Una aplicación que varia dinámicamente su comportamiento evidentemente está dirigida o/y es sensible al contexto para

poder adaptarse a él. Más adelante, entre 1997-1998, Hull [23] y Pascoe [26, 27, 13] definieron a las aplicaciones context-awareness por *"la habilidad que poseen los dispositivos computacionales de detectar, censar, interpretar y responder a los aspectos del ambiente del usuario".* Dey [14], en 1998, continua este camino y define la noción de las aplicaciones context-awareness como "*el uso del contexto para automatizar un sistema de software, modificar su interface y proveer la máxima flexibilidad en términos de servicios*". Salber [28] añade "*to be the ability to provide maximum flexibility of a computational service based on real-time sensing of context*", estas definiciones se centran en el conocimiento de determinados aspectos del contexto para brindar flexibilidad de servicios al usuario.
Por otro lado, Ryan [29], define a las aplicaciones context-awareness como "*aplicaciones que supervisan las entradas provenientes de los dispositivos de censado ambientales y permiten a los usuarios seleccionar un determinado contexto físico o lógico de acuerdo a sus intereses o sus actividades".* Esta definición es más restrictiva que las anteriores por que identifica el método por el cual las aplicaciones actúan sobre el contexto y sobre los intereses del usuario. Siguiendo esta línea, Brown [30] expresa, "*aplicaciones que automáticamente proporcionan información y/o reaccionan de acuerdo contexto actual del usuario que es detectado por los sensores".* Esta definición se centra en el contexto del usuario, es decir, toda la información relevante que influya sobre las intensiones del usuario; esta información provocan una reacción por parte de la aplicación que se ajusta a ese contexto. Este tipo de aplicaciones buscan ser proactivas, anticipándose a las acciones que puede llegar a realizar el usuario, un por ejemplo de esto podría ser, modificar la configuración de un "*layout*" grafico, para esto la aplicación deberá, por ejemplo, tener en cuenta aspectos contextuales como, la luminosidad del ambiente, el horario, la cantidad de batería restante (si es un dispositivo móvil), la actividad que está realizando el usuario en ese momento, etc.
Finalmente, Dey [31] expresa una aplicación context-awareness como: *"aquella que utiliza al contexto para proveer información relevante y servicios al usuario, donde la relevancia depende de la tarea que está llevando a cabo el usuario".*

4.2. Funcionalidades de la interfaz context-awareness

Para el entrenamiento de discapacitados audibles se considera las siguientes funcionalidades:

- Edición y recorrido de contextos para facilitara la composición de "Contextos", (Ej: Un contexto podría ser un paseo por el zoológico). El usuario podrá ver el contexto completo, con todos los objetos relacionados con el mismo. En este caso los objetos serán las distintas imágenes pertenecientes a dicho contexto (Ej: en el caso del zoológico, la jaula de los leones, los elefantes, etc). Cada una de estas imágenes será en realidad una "imagen de escena", ya que tendrán asociada una escena dinámica. Inicialmente el usuario podrá ver estas imágenes de escena en forma estática, mientras recorre el contexto. Cuando pase el mouse sobre alguna de estas imágenes de escenas, se reproducirá su nombre, el que podrá percibir con el dispositivo de estimulación, y podrá ver la simulación de los labios correspondiente.
- Cada contexto estará compuesto por al menos una imagen de escena.
- Cada imagen de escena tendrá asociada una escena dinámica que la representa
- El usuario podrá hacer clic con el mouse sobre cualquier imagen de escena que desee para acceder estas escenas dinámicas.
- Una escena dinámica, mostrará al elemento seleccionado en movimiento, y reproducirá los sonidos correspondientes al mismo. (Ej: la escena dinámica de los leones, podrá mostrar un león rugiendo, al mismo tiempo que se reproduce el sonido del rugido. Estos sonidos serán percibidos por el usuario a través del dispositivo de estimulación por vibraciones).

- Las imágenes de escena y las escenas correspondientes, podrían utilizarse como objetos visuales y audibles reutilizables en distintos contextos.
- Selección de contextos preprogramados, con diferentes niveles de dificultad:
  - El usuario tendrá la posibilidad de iniciarse en el uso del programa, con contextos previamente cargados.
  - Estos contextos estarán categorizados en distintos niveles de dificultad, que el usuario podrá seleccionar según sus preferencias y nivel de experticia alcanzado.
- Comunicación con una URL física ,como ya se dijo anteriormente, el usuario tendrá la posibilidad de descargar información de un objeto real desde una etiqueta RFID, a su celular o PDA, y percibir los sonidos de esta información con el dispositivo de estimulación por vibraciones
- Posibilidad de acceder a nuevos contextos existentes en la web a través de disponibilidad en el futuro de sitios con contextos para discapacitados auditivos, disponibles en internet
- Manejo personalizado de la información de los usuarios considerando que cada usuario se podrá registrar en el sistema, indicando su nivel de discapacidad, así como su nivel de experticia alcanzada en el uso del dispositivo,
- Historial asociado a cada usuario que contenga información relacionada con los entrenamientos realizados, incluyendo parámetros de seleccionados, y desempeño obtenido, como así también la información de los instructores que guían a cada usuario.

4.3. Diagrama de clases UML

A continuación se incluye un diagrama UML detallado en la figura 3, que representa en forma preliminar el modelo propuesto. El contexto del diagrama, representa al contexto que el usuario podrá visualizar y recorrer en forma global, hasta seleccionar una escena determinada. Las escenas estarán compuestas por al menos uno, o varios objetos (virtuales) de actuación, por ejemplo en el caso de una escena de autos, los objetos de actuación serán autos pero seguramente también habrá transeúntes, motos, etc. Dependiendo del contexto en que se desarrolle una determinada escena, serán los sonidos correspondientes a cada objeto de actuación, por ejemplo, en un contexto en el cual un automóvil está en un embotellamiento, el sonido asociado será el de la bocina
En cambio por ejemplo en otro contexto, en el que un animal se cruza repentinamente delante del camino de un auto, el sonido asociado será el de una frenada. Por otra parte, se hace una distinción en los distintos sonidos que pueden existir. Por un lado tendremos sonidos ambientales, que serán por ejemplo el maullido de un gato, el sonido del motor de un auto, el beep de un horno microondas, etc. Y por otro lado, tendremos los sonidos lingüísticos correspondientes a vocales y consonantes.
Además, existirán algunos objetos virtuales de actuación, para los que haya una correspondencia con un objeto real de actuación. El usuario podría estar en su casa recorriendo un contexto virtual, y viendo distintos objetos de actuación. Pero también podría ser que por ejemplo estuviera recorriendo un zoológico en donde los objetos serían reales, y sobre los cuales, si tienen etiquetas RFID, podría descargar información y sonidos a su celular o PDA, al mismo tiempo que podría percibir los sonidos con su dispositivo de estímulos por vibraciones

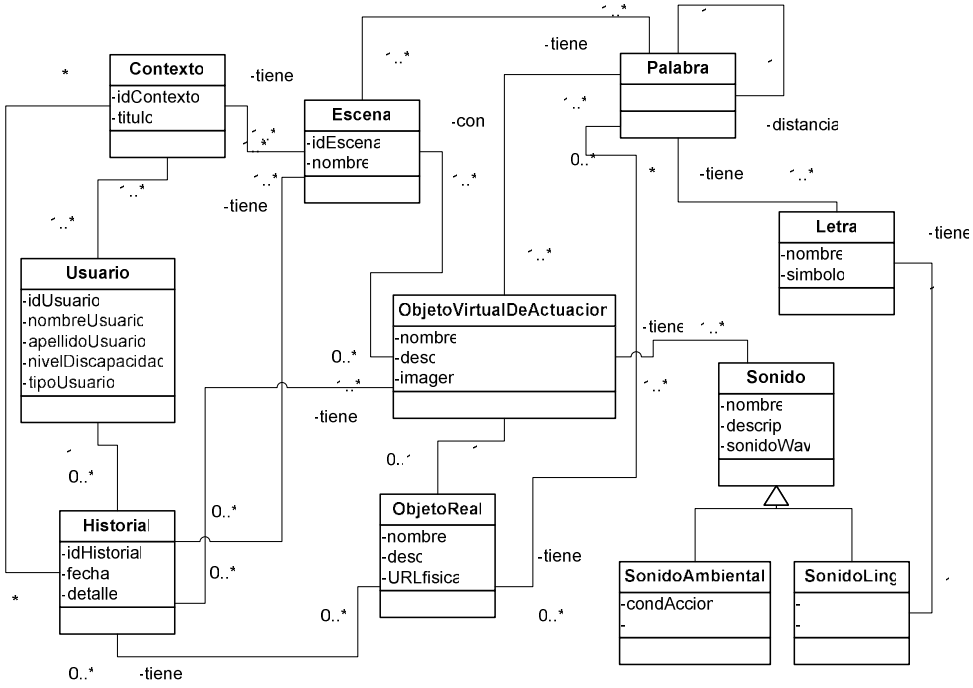


Figura 3: Diagrama de clases UML

.

4.4. Modelo Genérico del Entrenador

Considerando el modelo presentado por J Ierache, M Bruno [6] constituido por un mundo real, un mundo virtual y un mundo constructivo basado en [7] y los principios relacionados con la conformación del comportamiento descrito, se elabora el modelo de entrenador propuesto en la figura 4 "Modelo genérico del entrenador", el que se detalla a continuación:

- Dentro del mundo real se encontrarán los objetos reales de actuación. Suponiendo que el usuario está recorriendo un zoológico, y está viendo la jaula de las aves, los objetos reales de actuación en este caso serán las aves de la jaula (por ejemplo loros cacatúas, papagayos, etc).
- Dentro del mundo virtual, estarán los objetos virtuales de actuación, representados por información, imágenes digitales (por ejemplo, siguiendo el caso del paseo por el zoológico, fotos de loros, cacatúas, papagayos, etc).
- En el mundo constructivo, se encontrarán todas las escenas, tanto virtuales como reales. Este mundo constructivo, es el que permite integrar a los objetos dentro de escenas que el usuario puede luego recorrer.

Por otra parte, existen escenas reales que son implícitas, se producen cuando el usuario está en una determinada situación, en donde puede percibir sonidos generados por una escena real. (Por ejemplo, el usuario está frente a la jaula de las aves, y percibe los sonidos que estas producen, o está bajando la información y sonido de una URL física de una etiqueta RFID, a un dispositivo móvil, para enviarla al dispositivo de estimulación por vibraciones). Estas escenas reales, se ubican entre el mundo constructivo y el mundo real, ya que posee elementos de ambos. Las escenas virtuales, son explícitas. Están ubicadas entre el mundo virtual y el mundo constructivo. Son las escenas creadas a

propósito, para que el usuario pueda acceder a las mismas en la PC o en un dispositivo móvil, y percibir sus sonidos por medio del dispositivo de estimulación por vibraciones. Entre el mundo real y el mundo virtual se encuentran los sonidos, que podrán ser ambientales o lingüísticos, como ya se dijo anteriormente. Por último, el contexto se encuentra en la intersección de los tres mundos. El contexto contendrá distintas escenas, que podrán ser reales o virtuales, dependiendo de si el usuario está recorriendo un lugar físico y percibiendo sus sonidos, o si está accediendo a través de una PC o de un dispositivo móvil a alguno de los escenarios virtuales creados.

Figura 4: Modelo genérico del entrenador

5. CONCLUSIONES

La Concepción y Desarrollo de Software Sensible al Contexto [8], facilitó la elaboración de un modelo preliminar para el entrenamiento de discapacitados *con déficit auditivo severo* en diferentes contextos y escenas, como así también el empleo de interfaces hápticas y el potencial uso de URL físicas que permiten enriquecer las capacidades de explotación de los sonidos contextuales a través de interfaces hápticas.

La propuesta que conforma el contexto a partir de la interacción de situaciones del mundo real, constructivo y el virtual presentan un potencial para el crecimiento de capacidades para el entrenamiento del discapacitado.

Futuras actividades deberán centrarse en el desarrollo de un framework de simulación, la integración con los dispositivos hápticos, la integración de estos con dispositivos móviles.

6. REFERENCIAS

[1] Escudero J ; Campos L, Uso del dispositivo de estimulación vibro-táctil, Tesis FICCTE UM- 2004.

[2] Centro Argentino de Medios Alternativos de Comunicación (CAMAC). [en línea]. Disponible en Web: <www.centrocamac.com.ar >

[3] Fortier A, Rossi G, Gordillo S; Un enfoque orientado a objetos para software contex-aware. Facultad de Informática UNLP

[4] Roussos G; Enabling RFID, Computer March 2006

[5] Graber J ,et al ; Participatory Design with Proxies: Developing a Desktop-PDA System to Support People with Aphasia .2006

[6] Ierache J, Bruno M "Ontología para la Interacción de Agentes en un Hábitat inteligente, CACIC 2006.

[7] Holley C, Simulator Interoperability,TRW Systems, Journal of SAir Traficc Control ,1993

[8] Gordillo S, Rossi G Material del curso Concepción y Desarrollo de Software Sensible al Contexto Facultad de Informática UNLP. 2006

[9] Abowd Gregory D. and Dey Anind K. , "Towards a Better Understanding of Context and Context-Awareness", *Graphics, Visualization and Usability Center and College of Computing, Georgia Tech*, Atlanta, GA, USA, 1999.

[10] Dey Anind K., Salber Daniel, Abowd Gregory D., "The Conference Assistant: Combining Context-Awareness with Wearable Computing", GVU Center, College of Computing - Georgia Institute of Technology, Atlanta 1996.

[11] Schilit Bill and Theimer M., "Disseminating active map information to mobile hosts", IEEE Network, 1994, pages 22-32

.

[12] Brown P. J., Bovey J. D., and Chen X., " Context-aware applications: from the laboratory to the marketplace", *IEEE Personal Communications*, October 1997, pages 58-64.

[13] Ryan N., Pascoe J., Morse D., "Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant", *Computer Applications in Archaeology*, 1997.

[14] Dey A.K.," Context-Aware Computing: The Cyber Desk Project", *AAAI 1998 Spring Symposium on Intelligent Environments, Technical Report*, January 1998, pages 51-54.

[15] Brown P.J., "The Stick-e Document: a Framework for Creating Context-Aware Applications", *Electronic Publishing*, 1996, pages 259-272.

[16] Ward A., Jones A., and Hopper A., "A new location technique for the active office", 1997.

[17] Pascoe J., "Adding Generic Contextual Capabilities to Wearable Computers", *2nd International Symposium on Wearable Computers*, 1998, pages 92-99.

[18] Franklin D. and Flachsbart J., "All gadget and no representation makes jack a dull environment", *In Proceedings of AAAI 1998 Spring Symposium on Intelligent Environments*, 1998.

[19] Schilit B., Theimer M., "Disseminating Active Map Information to Mobile Hosts", *IEEE Network*, 1994, pages 22-32

[20] Brown M., "Supporting user mobility", *In IFIP World Conference on Mobile Communications*, 1996, pages 69–77.

[21] Cooperstock J., Tanikoshi K., Beirne G., Narine T., Buxton W., "Evolution of a Reactive Environment", 1995.

[22] Elrod S., Hall G., Costanza R., Dixon M., "Responsive Office Environments", CACM, 1993.

[23] Hull R., Neaves P., Bedford-Roberts J., "Towards Situated Computing", *1st International Symposium on Wearable Computers*, 1997, pages 146-153.

[24] Rekimoto J., Ayatsuka Y., Hayashi K., "Augment-able Reality: Situated Communication through Physical and Digital Spaces", *2nd International Symposium on Wearable Computers*, 1998.

[25] Kortuem Gerd, Segall Zary, and Bauer Martin, "Context-aware, adaptive wearable computers as remote interfaces to 'intelligent' environments", *In ISWC*, 1998, pages 58–65.

[26] Pascoe J., "Adding Generic Contextual Capabilities to Wearable Computers", *2nd International Symposium on Wearable Computers*, 1998.

[27] Pascoe J., Ryan N.S., Morse D.R., "Human-Computer-Giraffe Interaction – HCI in the Field", *Workshop on Human Computer Interaction with Mobile Devices,* 1998.

[28] Salber D., Dey A.K., Abowd G.D., "Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm", *Georgia Tech GVU Technical Report*, 1998.

[29] Ryan N., "Mobile Computing in a Fieldwork Environment: Metadata Elements", *Project working document,* 1997.

[30] Brown P.J., "Triggering Information by Context", *Personal Technologies*, 1998 pages 1-9.

[31] Dey A. K., "Providing architectural support for building contextaware applications", PhD thesis, Georgia Institute of Technology, 2000.

Optimizando la comunicación entre las Aplicaciones de Clientes y el motor del Modelo de Referencia de Workflow con Servicios Web

Daniel Riesco
Departamento de Informática, Universidad Nacional de San Luis
San Luis, C.P. 5700, Argentina
driesco@unsl.edu.ar

Paola Martellotto, Marcela Daniele
Departamento de Computación, Universidad Nacional de Río Cuarto
Río Cuarto, C.P. 5800, Argentina
paola@dc.exa.unrc.edu.ar, marcela@dc.exa.unrc.edu.ar

Abstract

This work proposes to optimize the communication between the Workflow Client Application Interface and the Workflow Engine of the Workflow Reference Model, defining a specification with Web Services. The Workflow Reference Model, developed by the Workflow Management Coalition (WfMC), shows an architecture that standardizes the development of Workflow applications, in order to allowing interoperability among them. It defines five interfaces and specifies a set of WAPIs (Workflow Aplication Programming Interfaces) that define the functions of these interfaces like calls to APIs in a third-generation language, forcing to know specifically where an application is located to be able to invoke it. The Web Services are autonomous and independent services that are offered by means of the Web. Its main benefit is that they allow that the applications more modular, facilitating the reusability in different platforms and programming languages. This specification with Web Services improves the communication of Client's Applications with the Workflow Engine, favoring the user since he doesn't need to know the exact location of the application to invoke, and to the applications because they can vary its location in the web without implying any change in its invocation.

Keywords: Workflow Reference Model, Workflow Client Application Interface, Web Service.

Resumen

Este trabajo propone optimizar la comunicación entre la Interfaz de Aplicaciones de Cliente y el motor Workflow del Modelo de Referencia de Workflow, definiendo una especificación con Servicios Web. El Modelo de Referencia de Workflow, desarrollado por la Workflow Management Coalition (WfMC), muestra una arquitectura que estandariza el desarrollo de aplicaciones Workflow, con el objeto de permitir la interoperabilidad entre los mismos. Define cinco interfaces y especifica un conjunto de WAPIs (Workflow Aplication Programming Interfaces) que definen las funciones de dichas interfaces como llamadas a APIs en un lenguaje de tercera generación, obligando a conocer específicamente dónde está ubicada una aplicación para poder invocarla. Los Servicios Web son servicios autónomos e independientes que se ofrecen mediante la web. Su principal beneficio es que permiten que las aplicaciones sean más modulares y desacopladas, facilitando su reutilización en distintas plataformas o lenguajes de programación. Esta especificación con Servicios Web mejora la comunicación de las Aplicaciones de Cliente con el motor Workflow, favoreciendo al usuario dado que no necesita conocer la ubicación exacta de la aplicación a invocar, y a las aplicaciones porque pueden variar su ubicación en la red sin implicar ningún cambio en su invocación.

Palabras Clave: Modelo de Referencia de Workflow, Interfaz de Aplicaciones de Cliente, Servicio Web.

1. Introducción

La globalización y los cambios de paradigmas empresariales, la evolución de las tecnologías de la información y la política liberal imperante hoy en el mundo, ubican a las organizaciones en el juego de la competitividad internacional. Las organizaciones se orientan a ser más horizontales hacia el enfoque de redes de procesos, los que deben ser diseñados de principio a fin empleando nuevas tecnologías. Un proceso de negocio es un conjunto de tareas relacionadas lógicamente que se ejecuta con la intención de obtener un resultado de negocio particular, el cual incluye recursos humanos así como los recursos materiales con el objetivo de producir un beneficio para la organización. El modelado de procesos de negocio permite visualizar las tareas, actividades y flujos, así como las unidades organizacionales diferentes que son afectadas por el proceso.
Los sistemas de Administración de Workflow se están desarrollando en las empresas como una nueva forma de organizar y administrar la información, ayudando a automatizar los procesos de negocio. Tales sistemas pueden predefinir un procedimiento de trabajo con la información relevante, el rol de los participantes en cada paso de trabajo, y la aplicación de software requerida para procesar cada paso. Cada nuevo caso se asigna automáticamente a los participantes en una secuencia concreta, la información se entrega a la gente que la necesita y las aplicaciones se ejecutan cuando es necesario.
Para poder tener cierto nivel de interoperabilidad entre los diversos productos de Workflow, es necesario definir un conjunto de interfaces y formatos para el intercambio de datos entre dichos componentes.
La Workflow Management Coalition (WfMC) ha desarrollado un Modelo de Referencia de Workflow [7] identificando las interfaces con las estructuras genéricas de las aplicaciones de Workflow, para permitir a los productos comunicarse a distintos niveles. En particular, para permitir la interacción de los usuarios con el motor de Workflow utiliza una Worklist, que es manejada por un administrador. La Interfaz de las Aplicaciones de Cliente es la encargada de manejar la interacción entre el motor de Workflow y el administrador de la Worklist. La WfMC ha especificado un conjunto de APIs (Aplication Programming Interfaces) para la administración de Workflows [4], las cuales están soportadas por los productos Workflow y se denominan Workflow Aplication Programming Interfaces (WAPIs).
El Modelo de Referencia de Workflow [7] fue desarrollado desde estructuras genéricas de aplicaciones de Workflow, identificando las interfaces con estas estructuras, para permitir a los productos comunicarse a distintos niveles. Todos los sistemas de Workflow contienen componentes genéricos que interactúan de forma predefinida.
Por otro lado, los Servicios Web son aplicaciones auto-contenidas, auto-descritas que pueden ser publicadas, localizadas e invocadas a través de la Web, sin la necesidad de conocer la ubicación exacta de los mismos [1]. Para la descripción de un Servicio Web se utiliza WSDL (Web Service Description Language) [4] [5] basado en XML [2]. Además, es necesario describir los mensajes entre las aplicaciones y el servicio web, y la forma en que los mismos serán transportados a través de la web. SOAP (Simple Object Access Protocol) [3] es el protocolo más conocido basado en mensajes, que es utilizado para describir la interacción de las aplicaciones con los web services. Por su parte, el protocolo de transporte más usado es HTTP (Hiper Text Transport Protocol). Y por último, es necesario registrar y localizar el servicio web, para lo cual se define un directorio de Servicios Web distribuido y basado en Web que permite que se listen, busquen y descubran. Por lo general este directorio es definido UDDI (Universal Description, Discovery and Integration) [6]. El uso de estos protocolos estándares permite lograr la interoperabilidad en ambientes heterogéneos, con independencia del Sistema Operativo, lenguaje de programación, etc.
Las implementaciones conocidas de la Interfaz de las Aplicaciones de Cliente [8] definen las funciones de las interfaces y su especificación, como llamadas a APIs en el lenguaje “C”, y para la invocación de una aplicación se requiere conocer específicamente su ubicación. En este trabajo se propone una

especificación de dicha Interfaz utilizando Web Services, con el objetivo de que el usuario del Workflow no necesite conocer la ubicación de la aplicación que desea invocar, y que cualquier aplicación pueda cambiar su ubicación en la red sin que esto implique ningún cambio en su invocación.

2. Estado del Arte

En [9] la WfMC presenta una propuesta de especificación de las Interfaces 2 y 4 basada en el uso de IDL y bindings con OLE como alternativas a las especificaciones existentes C y MIME. Los primeros trabajos de la WfMC (sobre WAPIs) se concentran en la definición de las funciones de las interfaces y su especificación como llamadas a APIs en el lenguaje "C". La especificación de la interoperabilidad se desarrolla subsecuentemente usando IDL para la especificación abstracta y bindings concretos basados en MIME, para usar vía Internet. En esta propuesta el manejo de las consultas WAPI se reemplaza por el uso de una colección de objetos OLE y se define un filtro para reemplazar el filtro WAPI. El IDL binding de los estándares de la OMG define un modelo de objetos que combina las interfaces 2, 4 y 5, y también direcciona el área de aplicaciones invocadas, donde las aplicaciones se asumen como objetos de negocio. Se define una especificación jFLOW usando la notación UML.
En [10] se presenta la especificación JointFlow, adoptada por 19 compañías en respuesta a OMG's Workflow Management Facility RFP. Esta especificación se basa en el trabajo de la Workflow Management Coalition y define las interfaces que soportan la interacción en tiempo de ejecución entre los componentes de Workflow, permite la interoperabilidad de los componentes de negocio a través de los dominios de negocio y permite el monitoreo de los procesos. La especificación JointFlow focaliza en definir un conjunto de interfaces que permiten la interoperabilidad de los componentes de Workflow, soporta el monitoreo y la asignación de recursos. La especificación JointFlow se utiliza para realizar aplicaciones Workflows distribuidas.
En [11] los autores presentan una especificación utilizando Grid Computing. En los WFMS (Workflow Management Systems) tradicionales los procesos son diseñados por herramientas de definición de procesos y ejecutados por el motor Workflow, en el mismo WFMS. Las tareas son desarrolladas por los usuarios finales o las aplicaciones. Los procesos solamente pueden ser ejecutados por motores que son accedidos por usuarios específicos y aplicaciones específicas. Por lo tanto, utilizar WFMS es restrictivo de las localizaciones. En esta propuesta las tareas son desarrolladas por los servicios Grid (los cuales se basan en un conjunto de interfaces estándar). Los servicios pueden ser accedidos por cualquier aplicación de acuerdo a estos estándares.
En [12] se introduce un agente de composición de Workflow, que es capaz de componer Workflow de Servicios Web, usar descripciones semánticas de los Servicios Web y encontrar Servicios Web para un Workflow. Se muestra cómo un Servicios Web se puede componer utilizando ontologías de Servicios Web semánticos, que se refiere a definir la semántica de un Servicios Web, es decir, su significado, más que sus parámetros de entrada y salida. Con esta tecnología se presenta un modelo para componer Workflow de Servicios Web. El modelo para la composición de Workflow describe qué ocurre cuando se crea una nueva instancia de Servicios Web. La instancia se publica en un directorio.
En [13] se propone encapsular la funcionalidad de las organizaciones en interfaces apropiadas y publicarlas como Servicios Web. Se espera que los Servicios Web puedan integrarse como parte de los procesos web. Pero esta integración es dificultosa dada la alta heterogeneidad, autonomía y distribución de la web. Una solución es el uso de ontologías. También es esencial para los Servicios Web soportar todas las fases del ciclo de vida de un proceso web. Se describe cómo el aplicar semántica a cada paso del ciclo de vida de un Proceso Web Semántico puede ayudar al rehúso, la integración y escalabilidad.

En [14] se presenta un lenguaje de Workflow que usa los Servicios Web como componentes, una arquitectura para un ambiente en tiempo de ejecución para este lenguaje, y aprovecha las ventajas de la utilización de esta clase de tecnología. La idea es desarrollar procesos de negocio formados por una composición de servicios que están disponibles en una red de computadoras. La tecnología Workflow se usa para coordinar las interacciones entre los Servicios Web. Los Servicios Web representan los pasos lógicos que componen un Workflow. El lenguaje propuesto AELCWS no soporta la interacción directa con las personas durante la ejecución de un proceso, de modo que la Interfaz de Aplicaciones de Cliente no esta definida ni implementada. La Interfaz de Invocación de las Aplicaciones sí es soportada por este lenguaje, a través de la invocación de los servicios que conforman la definición del proceso.

3. WorkFlows

Workflow es la automatización de los procesos de negocio donde los documentos, la información y tareas se pasan entre los participantes del sistema de acuerdo a un conjunto de reglas previamente establecidas [7].
Por su parte, un WFMS es un sistema que define, crea y administra la ejecución de Workflow a través del uso de software que se ejecuta sobre uno o más motores Workflow, los que interpretan la definición de los procesos, interactúan con los otros participantes del Workflow e invocan herramientas y aplicaciones [7].
Las tendencias imperantes en el mundo de los negocios obligan a las empresas a repensarse a sí mismas y rediseñar sus soportes tecnológicos, con el fin de: aumentar la productividad, mejorar la calidad, mejorar el servicio al cliente, reducir los costos y adaptarse a un entorno cambiante. En un WFMS:

- El trabajo no se extravía o detiene,
- Los gerentes pueden enfocarse en el personal y problemas de negocio,
- Los procedimientos son formalmente documentados,
- La mejor persona (o máquina) es asignada para hacer cada caso,
- El procesamiento paralelo es práctico.

El trabajo es realizado por el mejor participante, es decir, se distribuye el trabajo, y existe un supervisor quien puede influir en esta asignación automática. Además, el sistema necesita conocer qué trabajo espera ser asignado.

4. El Modelo de Referencia de la WfMC

El Modelo de Referencia de Workflow [7] identifica las interfaces con estructuras genéricas de aplicaciones, para permitir a los productos comunicarse a distintos niveles. Todos los sistemas de Workflow contienen componentes genéricos que interactúan de forma predefinida. La figura 1 muestra el Modelo de Referencia de Workflow propuesto por la WfMC.
En este modelo adoptado hay una separación entre los procesos y el control de la lógica de las actividades. Esta lógica esta dentro del Servicio de Representación de Workflow (Workflow Enactment Service). Esta separación permite la integración de las diversas herramientas con una aplicación particular.

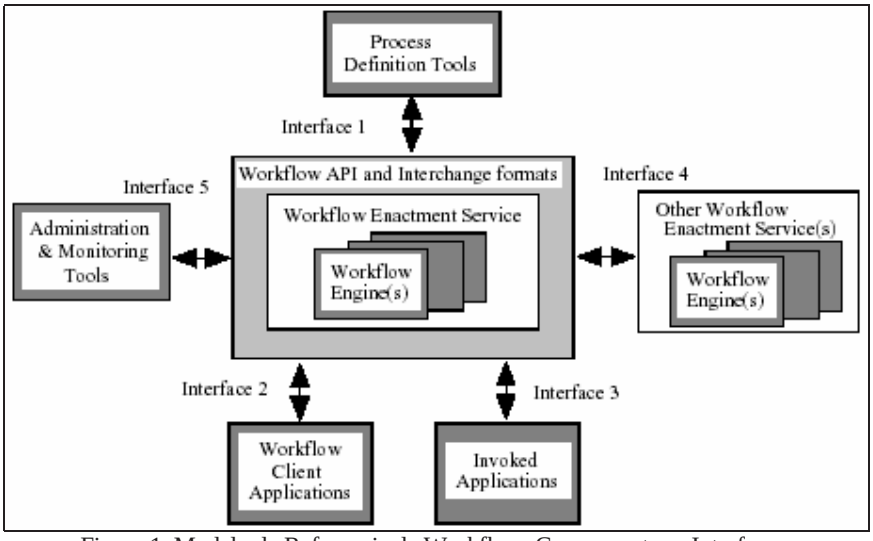


Figura 1. Modelo de Referencia de Workflow. Componentes e Interfaces.

El Servicio de Representación de Workflow interpreta la descripción de procesos y controla las diferentes instancias de los procesos, secuencia las actividades, adiciona ítems a la Lista de Trabajo (Worklist) de los usuarios, e invoca aplicaciones necesarias. La interacción del Servicio de Representación de Workflow con los recursos externos se da por una de las dos interfaces siguientes:

- La interfaz de las Aplicaciones de los Clientes, a través de la cual el motor de Workflow interactúa con el administrador de la Lista de Trabajo, responsable de organizar el trabajo por intermedio de un recurso de usuario.
- La interfaz de las Aplicaciones Invocadas, la cual le permite al motor de Workflow activar una herramienta para realizar una actividad particular. Esta interfaz podría estar basada en un servidor, es decir no existe la interacción con el usuario.

La Lista de Trabajo permite controlar la interacción con los usuarios. El motor deposita en la Lista de Trabajo los ítems ha ser ejecutados por cada usuario.
El Administrador de la Lista de Trabajo maneja la interacción entre los participantes del Workflow y el Servicio de Representación de Workflow, vía la Lista de Trabajo. Soporta un amplio rango de interacción con otras aplicaciones clientes.

5. Servicios Web

Los Servicios Web son aplicaciones auto-contenidas, auto-descritas que pueden ser publicadas, localizadas e invocadas a través de la Web. Una vez desarrolladas, otras aplicaciones (y otros Servicios Web) pueden descubrirlas e invocar el servicio dado [1].
El desarrollo y la programación de sistemas orientados a objetos o componentes han llevado a lo largo del tiempo a tener la necesidad de reutilizarlos en diferentes proyectos. Ya sean componentes desarrollados por uno mismo o componentes desarrollados por terceros. Hasta la existencia de los Servicios Web esta reutilización se limitaba a un lenguaje de programación o una plataforma en particular. Por lo tanto, el uso de los Servicios Web facilita la reutilización de una aplicación en distintas plataformas o lenguajes ya sea para uso personal en distintos proyectos, para comercializarlos o adquirir prestaciones de terceros.
De la misma forma que anteriormente se incluía en las aplicaciones referencias a otras librerías como ser DDLs, ahora se podrán referenciar funciones que se estarán ejecutando en otra computadora o servidor sin importar en qué están programadas ni en qué plataforma están corriendo.

Un Servicio Web se basa en las siguientes tecnologías:

- Para la descripción del servicio: un formato que describe la interfaz del componente (sus métodos y atributos) basado en XML [2]. Por lo general este formato es WSDL (Servicio Web Description Language) [4] [5]
- Para la representación de los mensajes: un protocolo de aplicación basado en mensajes que permite que una aplicación interactúe con el Servicios Web. Por lo general este protocolo es SOAP (Simple Object Access Protocol) [3]
- Para el transporte de los mensajes: un protocolo de transporte que se encargue de transportar los mensajes por Internet. Por lo general este protocolo de transporte es HTTP (Hiper Text Transport Protocol)
- Para el registro y la localización del servicio: un directorio de servicios web distribuido y basado en Web que permite que se listen, busquen y descubran este tipo de software. Por lo general este directorio es UDDI (Universal Description, Discovery and Integration) [6].

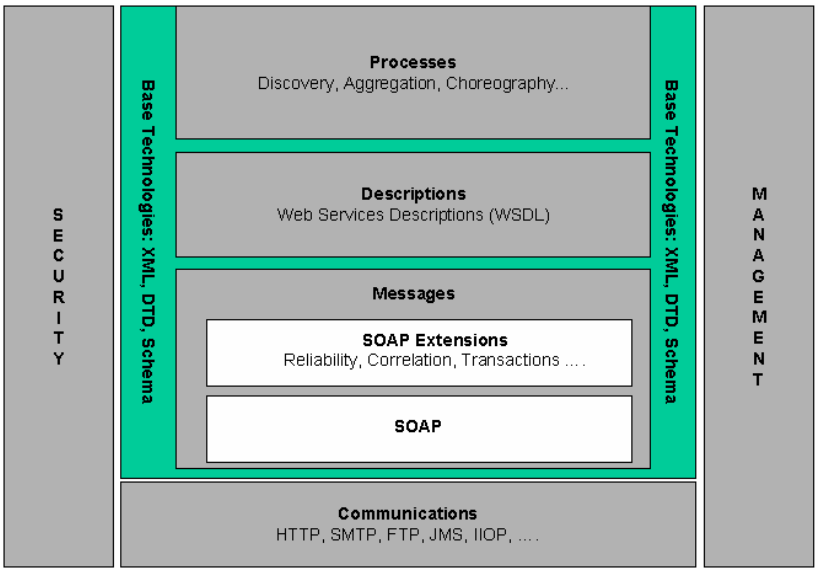


Figura 2. Arquitectura de los Servicios Web.

El uso de estos protocolos estándar es necesario para lograr la interoperabilidad en ambientes heterogéneos, con independencia del Sistema Operativo, el lenguaje de programación, etc.

WSDL [4] [5] es el lenguaje utilizado para describir un Servicio Web. WSDL describe un Servicio Web como un conjunto de puntos finales de comunicación (métodos) capaces de intercambiar mensajes. Es un archivo XML que describe el conjunto de métodos expuestos por un Servicio Web. Todo documento WSDL está compuesto por un elemento raíz llamado *definitions*, que a su vez está compuesto por los siguientes elementos:

- types: define qué tipos de datos serán transmitidos.
- message: define qué mensajes serán transmitidos.
- portType: define qué operaciones (funciones) serán soportadas.
- binding: establece cómo será transmitido el mensaje. Establece los detalles de la especificación SOAP.
- service: informa dónde se localiza el servicio.

WSDL se basa en el lenguaje XML (Extensible Markup Language) [2]. Este es el lenguaje utilizado para definir el formato de documentos o mensajes. XML comprende el uso de etiquetas denominadas tags que identifican los contenidos de un documento, y al hacerlo, los describen. Una etiqueta XML identifica información dentro de un documento, como así también la estructura de dicha información.

Los documentos XML poseen una estructura bien-formada y generalmente están asociados con un esquema (scheme) que especifica qué etiquetas están permitidas dentro de un documento, la estructura de esas etiquetas, y otras reglas relacionadas, tales como el tipo de dato que se espera dentro de una etiqueta.
Por otro lado, SOAP [3] es el protocolo utilizado en los Servicios Web para el transporte de los mensajes. Los mensajes SOAP están compuestos por un *tag* principal llamado *Envelope*, que está dividido en una cabecera o *Header* y un cuerpo o *Body*. Dentro del elemento Body estarán los elementos correspondientes al Web Method y además puede haber o no un elemento en común llamado *fault*, que indica que ha ocurrido un error y la razón de este. Por lo tanto, la definición de un Servicio Web implica, entonces, construir un documento WSDL que contenga los elementos mencionados anteriormente.

6. Especificación de la Interfaz de las Aplicaciones de Clientes con Servicios Web

Como ya se mencionó, la Interfaz de las Aplicaciones de Clientes permite la interacción entre las aplicaciones clientes y el motor de Workflow. Para sostener dicha interacción se utiliza una Lista de Trabajo, que almacena la información de las aplicaciones que se deben invocar y posee un administrador de la misma. La Lista de Trabajo puede contener ítems relacionados con diferentes instancias de un proceso o ítems de diferentes procesos. El administrador puede interactuar con diferentes motores.
Las APIs asociadas a esta interfaz proveen un nivel básico de funcionalidad para soportar la invocación de aplicaciones. Dichas APIs se pueden consultar en [8].
Las funciones de la Lista de Trabajo proveen información a los participantes del Workflow sobre los trabajos que ellos tienen asignados. Como lo describe el Modelo de Referencia de la WfMC, un proceso consiste de un conjunto de actividades conectadas de forma tal de controlar el secuenciamiento de la invocación a aplicaciones. Una actividad está asociada a una o más aplicaciones a ser invocadas y, en tiempo de ejecución, se asocia con la/s persona/s que han sido asignadas para realizar el trabajo.
Dependiendo de la implementación del Workflow, un participante de Workflow puede tener asignadas una o más piezas de trabajo al mismo tiempo. Cada pieza de trabajo se denomina Item de Trabajo (work item) y la colección de todos los Items de Trabajo asignados a un participante se denomina Lista de Trabajo (worklist) del participante de Workflow.
Cuando se define un proceso, la actividad es el constructor usado para definir una pieza de trabajo a realizar. En tiempo de ejecución, cuando la actividad esta lista para ser ejecutada y uno o más candidatos se asignan para realizar el trabajo, se crea un Item de Trabajo y se ubica en la Lista de Trabajo del participante. Aunque ambos representan una pieza de trabajo, existen en diferentes puntos en el tiempo, y puede haber más de un Item de Trabajo para una misma actividad. Para especificar la Interfaz de Aplicaciones de Cliente del Modelo de Referencia de Workflow con Servicios Web se procede a especificar cada una de las funciones definidas con WAPIs en dicho modelo. El caso de estudio que se presenta en este trabajo desarrolla la especificación de la función *WMOpenWorkList* , la cual provee información a los participantes del Workflow sobre los trabajos que tienen asignados.

6.1. La función WMOpenWorkList

La función *WMOpenWorkList* especifica y ejecuta una consulta para producir la Worklist que cumple con el criterio de filtro de la consulta.

El comando provee la capacidad de retornar la lista de work ítems asignados a una participante de Workflow particular o a un grupo de trabajo. El solicitante puede realizar la consulta a nombre de sí mismo o quizá esperar para conocer qué trabajos le han sido asignados a él particular o a un grupo de trabajo.
La consulta retornará la lista de todos los work items que coinciden con los valores específicos del atributo. El comando también puede retornar, opcionalmente, la cuenta total de los work ítems disponibles. La WAPI de esta función se puede consultar en [8].

6.2. Definición del Servicio Web para la función *WMOpenWorkList*

Para la función ***WMOpenWorkList***, uno de los elementos a definir son los tipos que usará el servicio. Estos tipos se declaran dentro del elemento <types>. Se utiliza un esquema XML para definir los tipos que son más complejos.

```
<types>
   <xs:schema
      …
      <xs:element name="psession_handle" type="Tpsession_handle"/>
      <xs:complexType  name="Tpsession_handle">
         <xs:sequence>
            <xs:element name="session_id" type="xs: String"/>
            <xs:element name="pprivate" type="xs: String"/>
         </xs:sequence>
      </xs:complexType>

      <xs:element name="pworklist_filter" type="Tpworklist_filter"/>
      <xs:complexType  name="Tpworklist_filter">
         <xs:sequence>
            <xs:element name="sqlString" type="xs: String"/>
            <xs:element name="attributeName" type="xs: String"/>
            <xs:element name="comparison" type="xs: int"/>
            <xs:element name="attributeValue" type="xs: boolean"/>
         </xs:sequence>
      </xs:complexType>

      <xs:element name="count_flag" type="Tcount_flag"/>
      <xs:complexType  name="Tcount_flag">
         <xs:sequence>
            <xs:element name="countFlag" type="xs: boolean"/>
         </xs:sequence>
      </xs:complexType>

      <xs:element name="pquery_handle" type="Tpquery_handle"/>
      <xs:complexType  name="Tpquery_handle">
         <xs:sequence>
            <xs:element name="ActivityDefinitionId " type="xs: String"/>
            <xs:element name="ActivityInstanceId" type="xs: String"/>
            <xs:element name="Id" type="xs: String"/>
            <xs:element name="Name" type="xs: String"/>
            <xs:element name="Participant" type="xs: String"/>
            <xs:element name="Priority" type="xs: int"/>
            <xs:element name="ProcessInstanceId" type="xs: int"/>
            <xs:element name="State" type="xs: String"/>
         </xs:sequence>
      </xs:complexType>
```

```
      <xs:element name="pcount" type="Tpcount"/>
      <xs:complexType  name="Tpcount">
         <xs:sequence>
            <xs:element name="count " type="xs: int"/>
         </xs:sequence>
      </xs:complexType>

      <xs:element name="WMInvalidSessionHandle" type="TWMInvalidSessionHandle"/>
      <xs:complexType  name="TWMInvalidSessionHandle">
         <xs:sequence>
            <xs:element name="excep" type="xs: String"/>
            <xs:element name="error" type="xs: String"/>
            <xs:element name="message" type="xs: String"/>
         </xs:sequence>
      </xs:complexType>

      <xs:element name="WMInvalidFilter" type="TWMInvalidFilter"/>
      <xs:complexType  name="TWMInvalidFilter">
         <xs:sequence>
            <xs:element name="excep" type="xs: String"/>
            <xs:element name="error" type="xs: String"/>
            <xs:element name="message" type="xs: String"/>
         </xs:sequence>
      </xs:complexType>
</xs:schema>
</types>
```

WSDL permite separar la descripción de la funcionalidad abstracta de un Servicio Web de los detalles concretos de cómo y dónde la funcionalidad es ofrecida. Una interfaz WSDL define la interfaz abstracta de un Servicio Web como un conjunto de operaciones abstractas.

```
<interface name = "WMOpenWorkListInterface" >
        <fault name = "WMOpenWorkList_InvalidSessionHandle"
                element = "wfms: WMInvalidSessionHandle"/>
        <fault name = "WMOpenWorkList_InvalidFilter"
                element = "wfms: WMInvalidFilter"/>

        <operation name="opWMOpenWorkList"
              pattern=http://www.w3.org/2004/03/wsdl/in-out>
              <input messageLabel="In" element="wfms: psession_handle" />
              <input messageLabel="In" element="wfms: pworklist_filter" />
              <input messageLabel="In" element="wfms: count_flag" />
              <output messageLabel="Out" element="wfms: pquery_handle" />
              <output messageLabel="Out" element="wfms: pcount" />
              <outfault ref="tns: WMOpenWorkList_InvalidSessionHandle" messageLabel="Out"/>
              <outfault ref="tns: WMOpenWorkList_InvalidFilter" messageLabel="Out"/>
        </operation>
</interface>
```

Otro elemento a especificar es el binding, que permite detallar cómo los mensajes pueden ser intercambiados. Especifica detalles del formato concreto de mensajes y del protocolo de transmisión para una interfase, y provee tales detalles para cualquier operación y falla en la interfaz.

```
<binding name="WMOpenWorkListSOAPBinding"
    interface="tns: WMOpenWorkListInterface"
    type=http://www.w3.org/2006/01/wsdl/soap
    wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP">

   <operation ref="tns: opWMOpenWorkList"
    wsoap:mep=…/>

    <fault ref="tns: WMOpenWorkList_ InvalidSessionHandle"
    wsoap:code="soap:Sender"/>

    <fault ref="tns: WMOpenWorkList_InvalidFilter"
    wsoap:code="soap:Sender"/>
</binding>
```

Finalmente, resta definir el servicio WMOpenWorkList. Esta definición implica especificar dónde el servicio puede ser accedido, mediante el uso del elemento service. Un servicio WSDL especifica una interfaz simple que soportará el servicio, y una lista de ubicación de puntos extremos (endpoints) donde ese servicio puede ser accedido.

```
<service name="WMOpenWorkListService"
    interface="tns: WMOpenWorkListInterface">
    <endpoint name="WMOpenWorkListEndpoint"
          Binding = "tns: WMOpenWorkListSOAPBinding"
          address = …/>
</service>
</description>
```

De esta manera quedan especificados los elementos types, interfaz, binding y service para la definición de la operación WMOpenWorkList, de la Interfaz de Aplicaciones de Clientes del Modelo de Referencia de Workflow utilizando SOAP y WSDL. Esto permite definir una nueva clase de aplicaciones que utilizan los servicios web distribuidos por la red, aprovechando las ventajas de la interoperabilidad.

7. Conclusiones

Los Sistemas de Administración de Workflow se desarrollan en las empresas con el objetivo de automatizar los procesos de trabajo en las oficinas. Tales sistemas pueden predefinir un procedimiento de trabajo con la información relevante, el rol de los participantes en cada paso de trabajo, y la aplicación de software requerida para procesar cada paso. Los Sistemas de Administración de Workflow realizan esta administración automáticamente y así, el trabajo se completa de una forma más eficiente.

La Workflow Management Coalition ha desarrollado un Modelo de Referencia de Workflow para permitir cierto nivel de interoperabilidad entre los diversos productos de Workflow. Las WAPIs para la administración de Workflows desarrolladas por la WfMC definen las funciones de las interfaces y su especificación como llamadas a APIs en el lenguaje “C”. Esta implementación requiere que la invocación de una aplicación implique conocer específicamente dónde está ubicada la misma. Los beneficios que otorgan los Servicios Web pueden ser aprovechados para mejorar esta implementación, facilitando la comunicación del Workflow con las aplicaciones. Con el uso de Servicios Web se logra

que el Workflow se comporte internamente de forma distribuida, es decir, que no necesite conocer dónde están las aplicaciones para invocarlas. Simplemente, requiere servicios y hay aplicaciones que le proveen dichos servicios.
Para especificar la Interfaz de Aplicaciones de Cliente del Modelo de Referencia de Workflow con Servicios Web se procede a especificar cada una de las funciones definidas con WAPIs en dicho modelo. En este trabajo se especificó la función *WMOpenWorkList* definiendo cada uno de los elementos que componen un Servicio Web. De manera análoga, las demás funciones de esta Interfaz pueden ser especificadas con Servicios Web, logrando una especificación independiente del lenguaje de programación, y de la plataforma subyacente.

8. Referencias Bibliográficas

[1] World Wide Web Consortium. Web Service Architecture. http://www.w3.org/TR/ws-arch/. Último acceso Mayo 2007.
[2] World Wide Web Consortium. Extensible Markup Language (XML) 1.0 (Second Edition). http://www.w3.org/TR/xml. Último acceso Mayo 2007.
[3] World Wide Web Consortium. SOAP Version 1.2 Part 1: Messaging Framework. http://www.w3.org/TR/soap12-part1/. Último acceso Mayo 2007.
[4] World Wide Web Consortium. Web Services Description Language (WSDL) Version 2.0 Part 0: Primer. http://www.w3.org/TR/wsdl20-primer. Último acceso Mayo 2007.
[5] World Wide Web Consortium. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. http://www.w3.org/TR/wsdl20. Último acceso Mayo 2007.
[6] OASIS. UDDI Version 3.0.2. http://uddi.org/pubs/uddi_v3.htm. Último acceso Mayo 2007.
[7] Workflow Management Coalition. The Workflow Reference Model. WfMC-TC00-1003. http://www.wfmc.org/standards/referencemodel.htm. Último acceso Mayo 2007.
[8] Workflow Management Coalition. Programming Interface 2&3 Specification. WfMC-TC-1009. V2.0. http://www.wfmc.org/standards/publicdocuments.htm. Último acceso Mayo 2007.
[9] Workflow Management Coalition. A Common Object Model Discussion Paper. WfMC-TC10-22. http://www.wfmc.org/standards/docs/TC-1022_commom_Object%20Model_Paper.pdf. Último acceso Mayo 2007.
[10] Schmidt M.T., Building Workflow Business Objects. IBM Software Group OOPSLA'98 Business Object Workshop IV.
[11] Yang M., Liang H., Xu B., S-WFMS: A service-based WFMS in Grid Environment. Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05). IEEE. 2005.
[12] Laukkanen M., Helin H.: Composing Workflows of Semantic Web Services. Workshop on Servicios Web and Agent-based Engineering. AAMAS'2003. Melbourne, Australia. 14/15 de Julio de 2003.
[13] Cardozo J., Shelt A., Introduction to Semantic Servicios Web and Web Process Composition. Publication of LSDIS. Large Scale Distributed Information Systems. University of Georgia. Computer Science Department.
[14] Hiane da S. Maciel L. A., Toshiro Yano E.: Uma Linguagem de WF Para Composicao de Web Services. XIX Simpósio Brasileiro de Engenharia de Software. Uberlandia, MG, Brasil. 2005.

Paginación + Discretización del Fixed Queries Trie

Carina Mabel Ruano, Ana Valeria Villegas y Norma Edith herrera
Departamento de Informática
Univ. Nacional de San Luis
Argentina
{nherrera,cmruano,anaville}@unsl.edu.ar

Resumen

La próxima generación de administradores de bases de datos deberá ser capaz de indexar datos multimedia y responder consultas de proximidad con tanta eficiencia como actualmente responden consultas de búsqueda exacta. Estas nuevas bases de datos se pueden modelar como un espacio métrico, sobre los cuales ya se han diseñando numerosas técnicas de indexación. El *Fixed Queries Trie (FQTrie)* es una de ellas y ha demostrado experimentalmente tener un buen desempeño.

En investigaciones anteriores hemos realizado trabajos en torno a mejorar la eficiencia del FQ-Trie desde varios tópicos diferentes: cantidad de evaluaciones de distancia,tiempo extra de CPU y tiempo de I/O. Con respecto a los dos primeros, hemos encontrado un método de discretización que logra mejorar la eficiencia del FQTrie. Con respecto al tiempo de I/O, hemos diseñado una técnica basada en el particionamiento del espacio que permite reducir el tiempo de I/O.

En este trabajo combinamos ambas técnicas a fin de lograr una implementación completa del FQTrie que sea eficiente tanto en memoria principal como en memoria secundaria.

1. Introducción

La problemática de búsquedas por similitud en bases de datos no tradicionales puede formalizarse por medio del modelo de *espacios métricos*. Un espacio métrico es un par $(\mathcal{X}, d)$, donde $\mathcal{X}$ es un conjunto de objetos y $d : \mathcal{X} \times \mathcal{X} \rightarrow R^+$ es una función de distancia que modela la similitud entre los elementos de $\mathcal{X}$. La función d cumple con las propiedades características de una función de distancia: $\forall x, y \in \mathcal{X},\ d(x, y) \geq 0$ (positividad), $\forall x, y \in \mathcal{X},\ d(x, y) = d(y, x)$ (simetría), $\forall x, y, z \in \mathcal{X},\ d(x, y) \leq d(x, z) + d(z, y)$(desigualdad triangular). La base de datos es un conjunto finito $\mathcal{U} \subseteq \mathcal{X}$.

Una de las consultas típicas que implica recuperar objetos similares de una base de datos es la *búsqueda por rango*, que denotaremos con $(q, r)_d$. Dado un elemento de consulta q, al que llamaremos *query* y un radio de tolerancia r, una búsqueda por rango consiste en recuperar aquellos objetos de la base de datos cuya distancia a q no sea mayor que r, es decir, $(q, r)_d = \{u \in \mathcal{U} : d(q, u) \leq r\}$.

Una forma trivial de resolver este tipo de búsquedas es examinando exhaustivamente la base de datos, es decir, comparando cada elemento de la base datos con la query. En general, esto resulta demasiado costoso en aplicaciones reales y no es posible realizarlo. Se han logrado avances importantes sobre espacios métricos en torno al concepto de construir un *índice*, es decir, una estructura de datos que permita reducir el tiempo necesario para resolver una consulta. En este tiempo T influyen tres factores; por un lado tenemos la cantidad de evaluaciones de la función de distancia d que se realizaron durante el proceso de búsqueda; por otro lado tenemos una cierta cantidad de operaciones adicionales que implican un tiempo extra de CPU; finalmente, tenemos un tiempo de I/O determinado por la cantidad de accesos a memoria secundaria, si es que fuera necesario; en símbolos:

T= #evaluaciones de $d\times$ complejidad(d) + tiempo extra de CPU + tiempo de I/O

Hay dos casos importantes a considerar: si el índice y los datos pueden ser mantenidos en memoria principal, o si es necesario utilizar memoria secundaria para los índices y/o datos. En el primer caso el primer término de T es el de mayor peso y, en consecuencia, el objetivo principal es reducir los cálculos de distancia realizados; en este sentido, un algoritmo se considera eficiente si puede contestar una consulta por similitud realizando un número pequeño de cálculos de distancia, sublineal respecto a la cantidad de elementos en la base de datos. Para los algoritmos en memoria secundaria, además de realizar pocos cálculos de distancia, se requiere que realicen pocos accesos a disco. Con respecto al tiempo extra de CPU, si bien suele ser el de menor peso en T, es importante reducirlo porque produce que en la práctica la búsqueda sea más rápida aún cuando estemos realizando la misma cantidad de evaluaciones de distancia y la misma cantidad de accesos a disco.

En [3] se presenta un desarrollo unificador de las soluciones existentes en la temática. En dicho trabajo se muestra que todos los enfoques para la construcción de índices en espacios métricos consisten en particionar el espacio en clases de equivalencia e indexar las clases de equivalencia. Luego, durante la búsqueda, usando el índice y la desigualdad triangular, se descartan algunas clases y se busca exhaustivamente en las restantes. La diferencia entre los distintos algoritmos de indexación radica en cómo construyen esta relación de equivalencia. Básicamente se pueden distinguir dos grupos:

Algoritmos basados en pivotes: estos algoritmos construyen la relación de equivalencia tomando la distancia de los elementos de la base a un conjunto preseleccionado de elementos denominados *pivotes*; sea $\{p_1, p_2, \ldots, p_k\}$ el conjunto de pivotes, dos elementos x e y son equivalentes si y solo si están a la misma distancia de todos los pivotes, es decir, $d(x, p_i) = d(y, p_i),\ \forall i = 1 \ldots k$. Durante la indexación, se seleccionan los k pivotes y se le asigna a cada elemento x de la base de datos el vector o firma $(d(x, p_1), d(x, p_2), \ldots, d(x, p_k))$. Ante una búsqueda $(q, r)_d$, se usa la

desigualdad triangular junto con los pivotes para filtrar elementos de la base de datos sin medir su distancia a la query q. Para ello se computa la distancia de q a cada uno de los pivotes p_i, y luego se descartan todos aquellos elementos a, tales que para algún pivote p_i se cumple que $\mid d(q,p_i) - d(a,p_i) \mid > r$. Los elementos no descartados forman parte de una *lista de candidatos*, que posteriormente se comparan directamente con la query q para decidir si forman o no parte de la respuesta.

Algoritmos basados en particiones compactas: en este caso la relación de equivalencia se construye teniendo en cuenta la cercanía de los elementos a un conjunto preseleccionado de elementos denominados *centros*; dos elementos son equivalentes si tienen al mismo centro c como su centro más cercano. Durante la indexación, seleccionan un conjunto de *centros* $\{c_1, c_2, \ldots, c_k\}$ y dividen el espacio asociando a cada centro c_i el conjunto de puntos que tiene a c_i como su centro más cercano. Existen muchos criterios posibles para descartar zonas durante una búsqueda. Los dos más populares son *criterio del hiperplano* y *criterio del radio de cobertura* [3].

Uno de los principales obstáculos en el diseño de buenas técnicas de indexación es lo que se conoce con el nombre de *maldición de la dimensionalidad*. El concepto de dimensionalidad está relacionado a la dificultad o facilidad de buscar en un determinado espacio métrico. La dimensión intrínseca de un espacio métrico se define en [3] como $\rho = \frac{\mu^2}{2\sigma^2}$, siendo μ y σ^2 la media y la varianza respectivamente de su histograma de distancias. Es decir que, a medida que la dimensionalidad intrínseca crece, la media aumenta y su varianza se reduce. Esto significa que el histograma de distancia se concentra más alrededor de su media, lo que influye negativamente en los algoritmos de indexación.

En investigaciones anteriores hemos trabajado en torno a mejorar la eficiencia del Fixed Queries Trie (FQTrie) [1], un índice basado en pivotes. En [4] se presenta un método de discretización, basado en los histogramas de distancias de los pivotes, que logra una implementación eficiente del FQTrie no sólo en términos de cantidad de evaluaciones de distancia de la función d, sino también en tiempo extra de CPU. En [6] se desarrolla una técnica de paginado que permite reducir el tiempo de I/O, que se basa en la idea de particionar la base de datos y agrupar en cada parte elementos similares.

En este artículo presentamos una implementación del FQTrie que combina ambas técnicas con el fin de lograr reducir las tres componentes que afectan el tiempo de resolución de una búsqueda por similitud: cantidad de evaluaciones de distancias, tiempo extra de CPU y cantidad de accesos a disco.

El artículo está organizado de la siguiente manera. Comenzamos en la sección 2 dando una breve reseña del índice FQTrie. Luego, en la sección 3, presentamos el método de discretización, el método de paginado y la forma de combinarlos. En la sección 4 exponemos la evaluación experimental de esta nueva propuesta y finalizamos en la sección 5 dando las conclusiones y el trabajo futuro.

2. Fixed Queries Trie

El *Fixed Queries Trie (FQTrie)* es una estructura basada en pivotes que fue presentado en [1] como una mejora al FQA [2] utilizando tablas lookup para mejorar los tiempos de búsquedas.

Habíamos visto en la introducción que todos los índices basados en pivotes asocian a cada elemento x de la base de datos, la firma del mismo $(d(x,p_1), d(x,p_2), \ldots, d(x,p_k))$, donde $\{p_1, p_2, \ldots, p_k\}$ es el conjunto de pivotes. La diferencia entre los distintos algoritmos basados en pivotes radica en cómo buscan dentro de ese conjunto de firmas. En el caso del FQTrie las firmas de los elementos son consideradas como cadenas de caracteres de longitud fija y se utiliza un *Árbol Digital* o *Trie* [5] para indexar las firmas de todos los elementos de la base de datos.

El FQTrie representa cada firma $(d(x,p_1),\ d(x,p_2),\ldots,d(x,p_k))$, haciendo uso de una función de discretización. Dicha función mapea los números reales positivos devueltos por la función d en valores discretos de tamaño b_p bits. Este valor discreto depende del valor devuelto por $d(x,p_i)$ y del pivote p_i. Formalmente la función de discretización se define como sigue:

$$\delta : \mathbb{R}^+ \times \mathbb{K} \rightarrow \{0,\ldots,2^{b_p}-1\}$$

De esta forma, para un elemento $x \in \mathbb{X}$ la firma de x se obtiene como la concatenación de la discretización de las distancias del objeto a cada pivote, en símbolos:

$$\delta^*(x) = \delta_{p_1}(d(x,p_1)) \centerdot \delta_{p_2}(d(x,p_2)) \centerdot \ldots \centerdot \delta_{p_k}(d(x,p_k))$$

Podemos extender la función δ_p a intervalos obteniendo así un conjunto de firmas individuales:

$$\delta_p([r_1,r_2]) = \{\delta_p(r)/r \in [r_1,r_2]\}$$

Formalmente δ_p está definida para números reales positivos, Podemos extender esta definición a todo el dominio de los números reales de la siguiente manera: $\delta([r_1,r_2]) = \delta([0,r_2])$ si $r_1 < 0$.

Ante una búsqueda $(q,r)_d$ se calcula la firma de la query definida de la siguiente manera:

$$\delta^*((q,r)_d) = \{\delta_{p_1}([d(q,p_1)-r, d(q,p_1)+r])\} \centerdot \ldots \centerdot \{\delta_{p_k}([d(q,p_k)-r, d(q,p_k)+r])\}$$

Es decir que $\delta^*((q,r)_d)$ es un conjunto de firmas que indica cuál debería ser la firma de un elemento para que sea candidato a formar parte de la respuesta de la query. Con estas definiciones, usando la desigualdad triangular, se puede demostrar que si x satisface la búsqueda $(q,r)_d$, entonces $\delta^*(x) \in \delta^*((q,r)_d)$. Luego, la lista de candidatos se puede definir como: $\{x/\delta^*(x) \in \delta^*((q,r)_d)\}$

Denotaremos con $\mathcal{U}^*$ al conjunto de firmas de la base de datos $\mathcal{U}$: $\mathcal{U}^* = \{\delta^*(x)/x \in \mathcal{U}\}$. Con las notaciones dadas, computar la lista de candidatos es equivalente a realizar la intersección $\mathcal{U}^* \cap \delta^*((q,r)_d)$. Como ya lo mencionáramos, en el FQTrie cada firma es considerada como una secuencia de caracteres y se utiliza un Árbol Digital o Trie para representar $\mathcal{U}^*$. Luego, la intersección anterior se calcula usando el Trie.

Notar que existe una cantidad exponencial de firmas en $\delta^*((q,r)_d)$. Las firmas se obtienen como concatenación ordenada de firmas respecto de cada pivote. Esto significa que, si cada pivote produce v_{p_i} firmas, entonces $|\delta^*((q,r)_d)| = \prod_{i=1}^{k} v_{p_i}$. Por ejemplo, si tenemos 32 pivotes, y para una búsqueda cada pivote produce 2 firmas, entonces $|\delta^*((q,r)_d)| = 2^{32}$. En consecuencia, no es viable calcular explícitamente el conjunto de firmas. En su lugar, se utiliza una representación implícita: *tablas lookup* [1].

Buscar una cadena en un Trie toma tiempo proporcional a la cantidad de caracteres en ella, independientemente de la cantidad de elementos contenidos en el conjunto. Dada una cadena, los caracteres que la conforman son los que direccionan la búsqueda en el Trie. Para el caso del FQTrie, la búsqueda se realiza con la asistencia de la *tabla lookup* la cual contiene el conjunto de firmas de la búsqueda ($\delta^*(q,r)$), permitiendo búsquedas de múltiples cadenas en el mismo recorrido.

3. Paginación y Discretización del FQTrie

En investigaciones anteriores hemos trabajado en torno a mejorar la eficiencia del FQTrie. En [4] se presenta un método de discretización, basado en los histogramas de distancias de los pivotes, que logra una implementación eficiente del FQTrie no sólo en términos de cantidad de evaluaciones de distancia, sino también en tiempo extra de CPU. En [6] se desarrolla una técnica de paginado que

permite reducir el tiempo de I/O, que se basa en la idea de particionar la base de datos y agrupar en cada parte elementos similares. En este trabajo combinamos ambas técnicas con el fin de lograr una implementación del FQTrie que sea eficiente tanto en memoria principal como en memoria secundaria.

Comenzaremos explicando la técnica de paginado, luego presentamos las funciones de discretización y finalmente detallamos la forma de combinar ambas técnicas.

3.1. Técnica de Paginado para el FQTrie

En [6] se presenta una técnica que permite manejar espacios métricos cuyo índice completo y/o datos exceda la capacidad de la memoria principal. Para ello, en lugar de usar el enfoque tradicional de modificar la estructura para que sea eficiente en memoria secundaria, se particiona el espacio de manera tal que cada una de las partes entre en memoria principal, las que posteriormente se indexan en forma separada. Luego, una búsqueda se resuelve buscando en cada parte, lo que puede ser hecho en memoria principal.

La técnica presentada en dicho artículo trata de agrupar objetos que sean similares dentro de cada parte con el objetivo de que, ante una consulta, una parte o contiene muchos elementos relevantes para la query o no contiene ninguno. De esta manera, se intenta amortizar los costos de accesos a las partes que residen en disco.

Una consulta $(q, r)_d$ se resuelve realizando los siguientes pasos con cada una de las partes que conforman la base de datos:

- se carga el índice de la parte.
- se busca en él para obtener la lista de elementos candidatos.
- si la lista de candidatos no es vacía, se carga la parte correspondiente en memoria principal, para poder comparar cada elemento de la lista contra la query q.

La técnica de particionamiento diseñada se basa en la distancia LCS(Longest Common Subsequence) que calcula la longitud de la máxima subsecuencia común entre dos cadenas. Esta técnica, que denotaremos con PLCS, detecta grupos de elementos parecidos a partir de los cuales genera la partición.

En [6] se estudia el comportamiento de PLCS sobre diccionarios de palabras con la función de distancia de edición y sobre documentos de texto con la función de distancia coseno. Allí se concluye que PLCS es competitiva cuando se utiliza como espacio métrico diccionarios de palabras, logrando disminuir la cantidad de accesos a disco en un $21\,\%$ respecto de un particionado totalmente aleatorio. También se demuestra que es eficiente disminuyendo cantidad de accesos a disco requeridos ante una consulta, respecto de no particionar la base de datos y dejar el manejo al sistema operativo. El comportamiento ha sido diferente cuando el espacio métrico utilizado son documentos de texto y los resultados no han sido tan alentadores respecto de no particionar la base de datos. En el mencionado artículo se concluye que, si los elementos de la base de datos son mayores que el tamaño de un página de disco, no es bueno particionar el espacio métrico con PLCS ya que eso implica almacenar punteros en cada parte y los resultados han mostrado que, para estos casos, no es posible obtener ventaja de dividir el espacio métrico en partes, aún cuando la división se realiza en forma controlada.

3.2. Funciones de Discretización

La principal ventaja de la función de discretización se centra en el grado de libertad en cuanto al uso de memoria disponible ya que podemos decidir cuántos bits asignar a cada pivote y determinar así

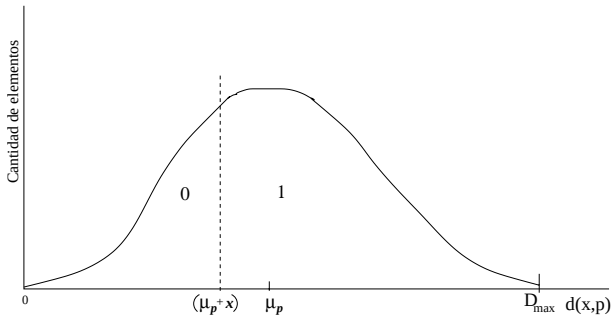


Figura 1: *Partición provocada por la función de discretización* δ_{μ_p} *con* $x < 0$.

el tamaño de la firma de cada objeto. Además permite establecer un balance entre el poder de filtrado y el espacio utilizado ya que, a medida aumenta la cantidad de bits asignados a un pivote, aumenta la presición pero disminuye la cantidad de pivotes que pueden ser contenidos en la misma cantidad de espacio.

En [2] y [4] se introducen varias funciones de discretización, entre la que se destacan:

- **Partes Iguales**: esta técnica consiste en dividir el espacio en partes de igual tamaño. Sea $P = \{p_1, p_2, \cdots, p_k\}$ el conjunto de pivotes. Para cada p_i se calcula $D_{max} = max_{u \in (\mathcal{U}-P)}\{d(p_i, u)\}$ y $D_{min} = min_{u \in (\mathcal{U}-P)}\{d(p_i, u)\}$. Luego, el rango $D_{max} - D_{min}$ es dividido en 2^{b_i} partes iguales, asociando a cada número $v \in \{0 \ldots 2^{b_i} - 1\}$ el intervalo $[D_{min} + v(D_{max} - D_{min})/2^{b_s}, D_{min} + (v+1)(D_{max} - D_{min})/2^{b_s}]$. Si bién esta técnica asegura que todas las partes son del mismo tamaño, no asegura que la cantidad de elementos en cada parte sea la misma.

- **Cantidades Iguales** : esta técnica divide el espacio intentando dejar la misma cantidad de elementos en cada parte. Por cada pivote se determina los $b_i - 1$ cuantiles uniformes que dividen el conjunto de valores de distancias en b_i subconjuntos de la misma cardinalidad. Luego se asigna un cuantil a cada valor entre 0 y $b_i - 1$. Esta técnica asegura que en cada intervalo existen exactamente $n/2^{b_i}$ objetos.

- **Media** : esta función divide el histograma local de un pivote p en dos partes, utilizando como límite divisor $\mu_p + x$, donde μ_p es la media del histograma y x es un número real. Luego, asigna 0 a todos aquellos valores que se ubiquen en el histograma a izquierda del límite divisor y 1 a los que se ubiquen a derecha (ver figura 1). El objetivo de esta función es divir la zona de mayor concentración de elementos.

De acuerdo a los resultados expuestos en [4], se concluye que la función *media* resulta ser la mejor elección si el histograma tiene forma de campana, siendo más competitiva que *partes iguales* y *cantidades iguales*. Si el histograma no tiene forma de campana, se realiza una adaptación de la función *media* a fin de que la misma divida la zona de mayor concentración de elementos.

3.3. Combinando Ambos Métodos

En este trabajo el objetivo es combinar ambas técnicas a fin de lograr un implementación del FQTrie que sea eficiente en términos de las tres componentes que afectan el tiempo de resolución de una consulta: cantidad de evaluaciones de la función de distancia d, cantidad de accesos a disco y tiempo extra de CPU.

Para ello, paginamos usando el método PLCS y construimos el índice de cada parte usando las funciones de discretización al momento de construir las firmas de los elementos. La figura 2 muestra

```
PLCS+Discretización(in 𝒰 , in δ, in B)
     1. ParticionarLCS(𝒰 ,𝒫𝒰)
     2. ℱ = ∅
     3. Para cada 𝒰_i ∈ 𝒫𝒰 hacer
     4.    Cargar 𝒰_i en memoria principal
     5.    Obtener 𝒰_i* = {δ*(x) : x ∈ 𝒰_i}
     6.    Indexar (𝒰_i*, FQTrie)
     7.    Si (size(ℱ) + size (FQTrie) >B )
     8.       grabar(ℱ)
     9.       ℱ = FQTrie
     10.   sino ℱ = ℱ ∪ FQTrie
     11. Fin Para
```

Figura 2: *Algoritmo que combina particionado LCS con funciones de discretización*

el pseudocódigo del algoritmo que permite combinar el método de paginación con el método de discretización. Como parámetros de entrada recibe la base de datos $\mathcal{U}$, la función de discretización δ y el tamaño de la página de disco B. El proceso ParticionarLCS es el que aplica la técnica de particionado a $\mathcal{U}$, obteniendo la partición del mismo $\mathcal{PU}$; este proceso se encarga además de grabar cada parte obtenida en una página disco. Dado que generalmente el FQTrie de una parte es pequeño, se agrupan varios de estos índices por página de disco. En el algoritmo se usa la variable $\mathcal{F}$, que mantiene el conjunto de índices creados y aún no grabados. Cuando el tamaño de $\mathcal{F}$ se adecúa a B, se graba el grupo corriente de índices y se comienza la creación de un nuevo grupo de índices.

Para las búsquedas no hay modificaciones respecto de lo explicado en la sección 3.1

Cabe señalar que, dado que el método de paginado PLCS es eficiente sólo para espacios cuyos objetos son, en tamaño, menor que el tamaño de una página de disco, la propuesta que aquí presentamos sólo es aplicable a espacios métricos con esta característica.

4. Evaluación Experimental

La evaluación de la técnica de particionado se realizó usando como espacio métrico diccionarios de palabras con la función de distancia de edición. Esta función es discreta y calcula la mínima cantidad de caracteres que hay que agregar, cambiar y/o eliminar a una palabra para obtener otra.

Se usaron en total 3 diccionarios: Español de $86{,}061$ palabras, Inglés de $69{,}069$ palabras y Francés de $138{,}257$ palabras. Cada uno de estos diccionarios fue paginado y luego indexado con firmas de tamaño 1, 2 y 4 bytes, y usando las funciones de discretización *media*, *cantidades iguales* y *partes iguales*.

Para cada diccionario se eligieron al azar 500 palabras las que fueron utilizadas en todos los experimentos. Para cada palabra de este grupo, se realizaron búsquedas por rango usando como radio de búsqueda r los valores $1, 2, 3$ y 4. Para cada búsqueda se contabilizó la cantidad de evaluaciones de distancias realizadas, la cantidad de páginas de disco accedidas y el tiempo total usado para resolver la consulta. En cada gráfica se muestran los promedios sobre las 500 búsquedas realizadas.

A fin de evaluar la eficiencia de combinar los métodos de discretización y paginado, los resultados obtenidos fueron comparados con los obtenidos paginando pero sin discretizar. Por cuestiones de espacio sólo mostramos las gráficas que consideramos más representativas; las restantes se encuentran disponibles para quien así lo requiera.

La figura 3 muestra los resultados para el diccionario Español usando firmas de 1 byte. En la parte superior se ha graficado el promedio de cantidad de evaluaciones de distancias (izquierda) y de

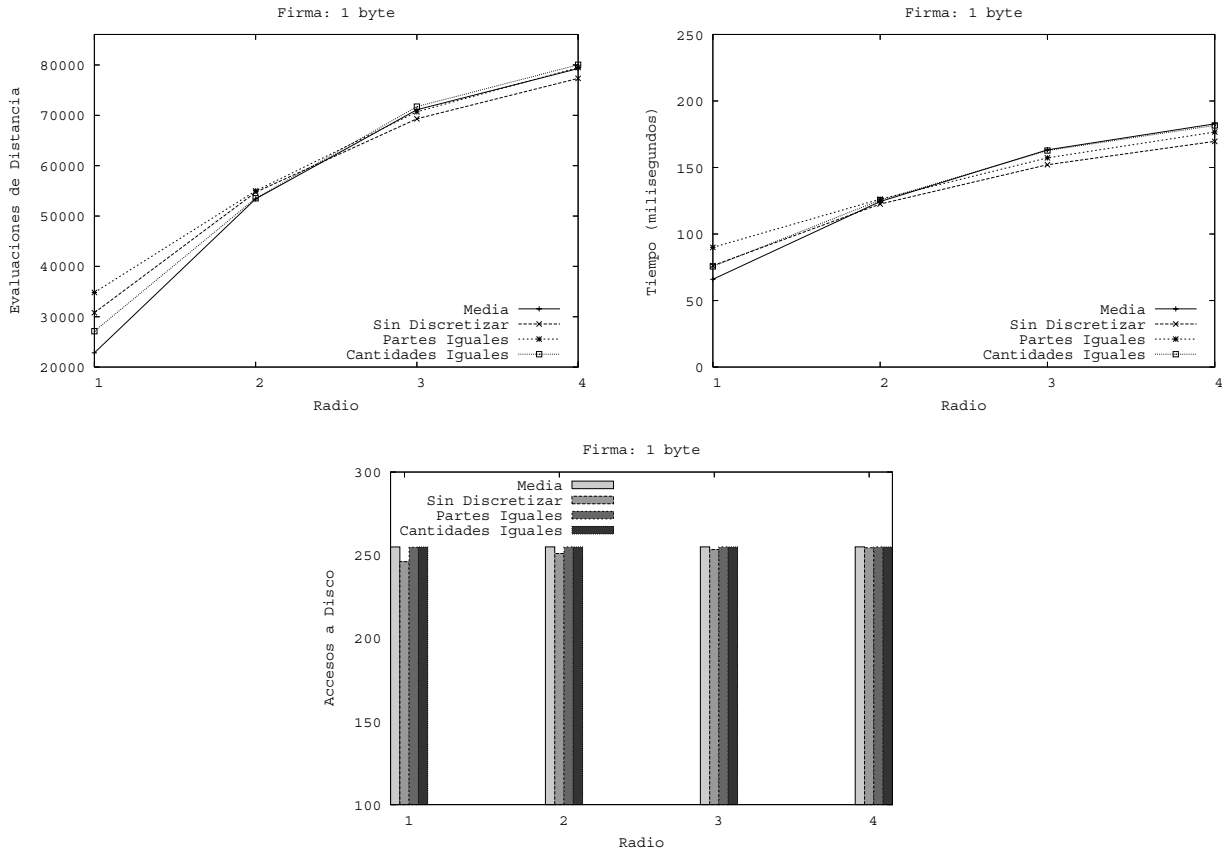


Figura 3: *Diccionario Español, tamaño de firma 1 byte*

tiempos totales de búsqueda (derecha); en la parte inferior se ha graficado el promedio de páginas accedidas durante el proceso de búsqueda. Se puede observar que, para búsquedas de alta selectividad (radio 1 y 2), PLCS con *media* y PLCS con *cantidades iguales* son más competitivas, en ese orden, que la opción de particionar con PLCS pero sin discretizar, logrando menor cantidad de evaluaciones de distancias y mejores tiempos de búsqueda. Notar que para radio 1, PLCS sin discretizar realiza menos accesos a disco que las restantes opciones, pero aún así no logra superarlas en tiempo. La razón de este comportamiento es que PLCS sin discretizar realiza aproximadamente un $30\,\%$ más de evaluaciones de distancias que PLCS con *media* y sólo un $5\,\%$ menos de acceso a disco; por lo tanto, en el tiempo total, termina pesando más esa cantidad extra de evaluaciones de distancias realizadas.

Esta situación cambia cuando consideramos búsquedas de baja selectividad (radios 3 y 4). Si bien todas las posibilidades consideradas obtienen resultados similares, PLCS sin discretizar y PLCS con *partes iguales* logran un desempeño levemente superior que las restantes. PLCS sin discretizar realiza un $5\,\%$ menos de evaluaciones de distancia que las restantes y tarda 10 milisegundos menos que PLCS con *partes iguales*, la que le sigue en eficencia.

La figura 4 muestra los resultados para tamaño de firma de 2 bytes. Se puede observar que, como era de esperar, todas las opciones logran mejorar su rendimiento dado que al tener mayor espacio se pueden usar mayor cantidad de pivotes lo que mejora la performance de cualquier índice basado en pivotes. Pero la que mayor beneficio obtiene es PLCS sin discretizar. Para búsquedas de radio 1, PLCS sin discretizar logra acercarse en cantidad de evaluaciones de distancia e igualar en tiempo a PLCS con *media*. Intuimos que la razón de esto es que PLCS sin discretizar, para radio 1, realiza menos accesos a disco que PLCS con *media* lo que redunda en beneficio del tiempo total de búsqueda.

Para búsquedas de baja selectividad, nuevamente PLCS sin discretizar y PLCS con *partes iguales*

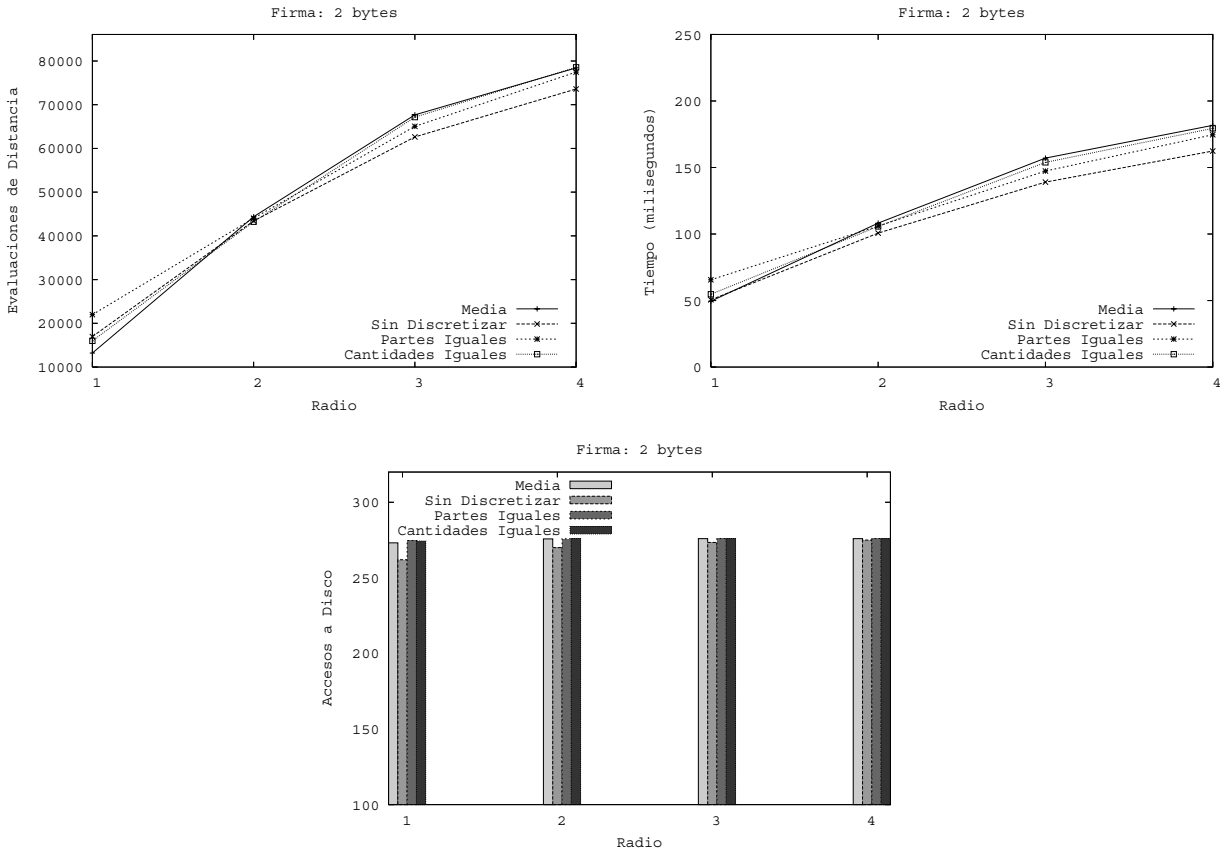


Figura 4: *Diccionario Español, tamaño de firma 2 bytes*

son las más competitivas. Con respecto a la cantidad de accesos a disco, puede observarse que en todos los casos se realizan una cantidad similar de accesos. La mayor diferencia se da en búsquedas de radio 1 siendo en este caso PLCS sin discretizar la que realiza menor cantidad de accesos a disco.

Finalmente, en la figura 5 se grafican los resultados para firmas de 4 bytes. En este caso, PLCS sin discretizar es la más competitiva logrando superar a PLCS con *media* para búsquedas de radio 1. Cabe destacar que para lograr este resultado, PLCS sin discretizar debió cuadruplicar el espacio pasando de firmas de 1 byte a firmas de 4 bytes.

Un punto importante para señalar es el efecto del tamaño de firma sobre la cantidad de accesos a disco. Aumentar el tamaño de firma beneficia a los índices porque permite aumentar la cantidad de pivotes lo que mejora el el poder de filtrado. Pero aumentar el tamaño de firma también provoca que los índices sean más grandes y se necesite una mayor cantidad de páginas para contenerlos. Dado que una búsqueda necesita acceder a todos los índices, aumentar el tamaño de firma tendrá como consecuencia un incremento en la cantidad de accesos a disco. Por ejemplo, para búsquedas de radio 1 y firmas de 1 byte las técnicas planteadas utilizan alrededor de 250 accesos a disco; para búsquedas de radio 1 y firmas de 4 bytes utilizan alrededor de 300 accesos a disco. La figura 6 ilustra la observación anterior para búsquedas de radio 1 y 3. Esta diferencia en cantidad de accesos a disco, se ve amortizada por la reducción en cantidad de evaluaciones de distancia lo que provoca finalmente menores tiempo de búsqueda.

Las observaciones hechas en el párrafo anterior nos indican que no se puede aumentar indiscriminadamente el tamaño de firma porque, aún cuando el índice de cada parte entre en memoria principal, en algún punto la cantidad de accesos a disco pesará más que los beneficios obtenidos por la reducción en cantidad de evaluaciones de distancias y perjudicaremos los tiempos de búsqueda.

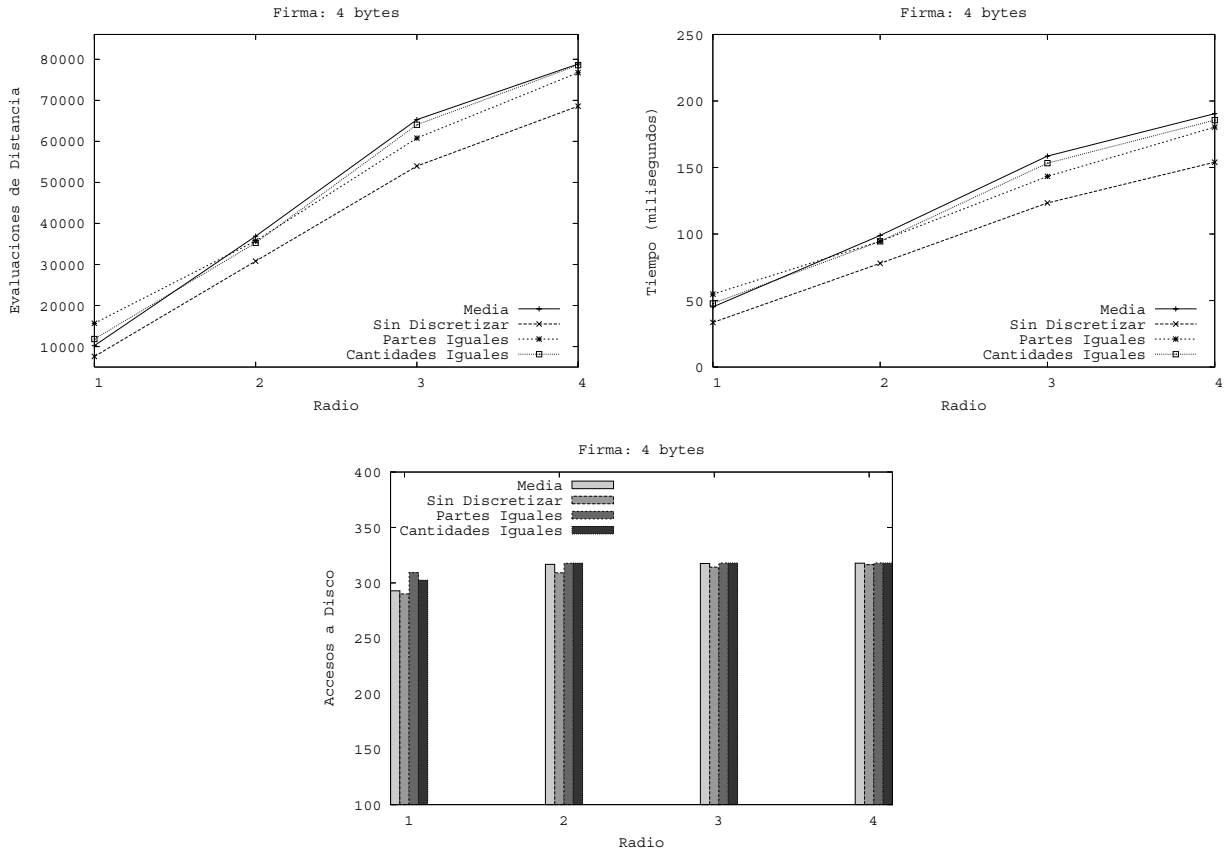


Figura 5: *Diccionario Español, tamaño de firma 4 bytes*

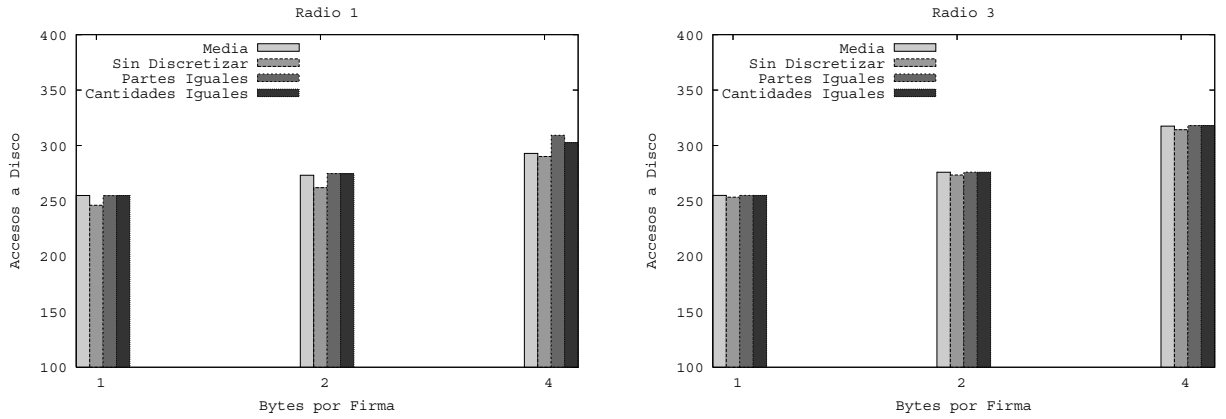


Figura 6: *Diccionario Español, efecto del tamaño de firma sobre la cantidad de accesos a disco*

Los resultados sobre los restantes diccionarios fueron similares y, en consecuencia, se obtienen las mismas conclusiones que las detalladas para el diccionario Español. Las figuras 7 y 8 muestran los resultados para el diccionario Inglés y el dicionario Francés con tamaño de firma de 1 byte.

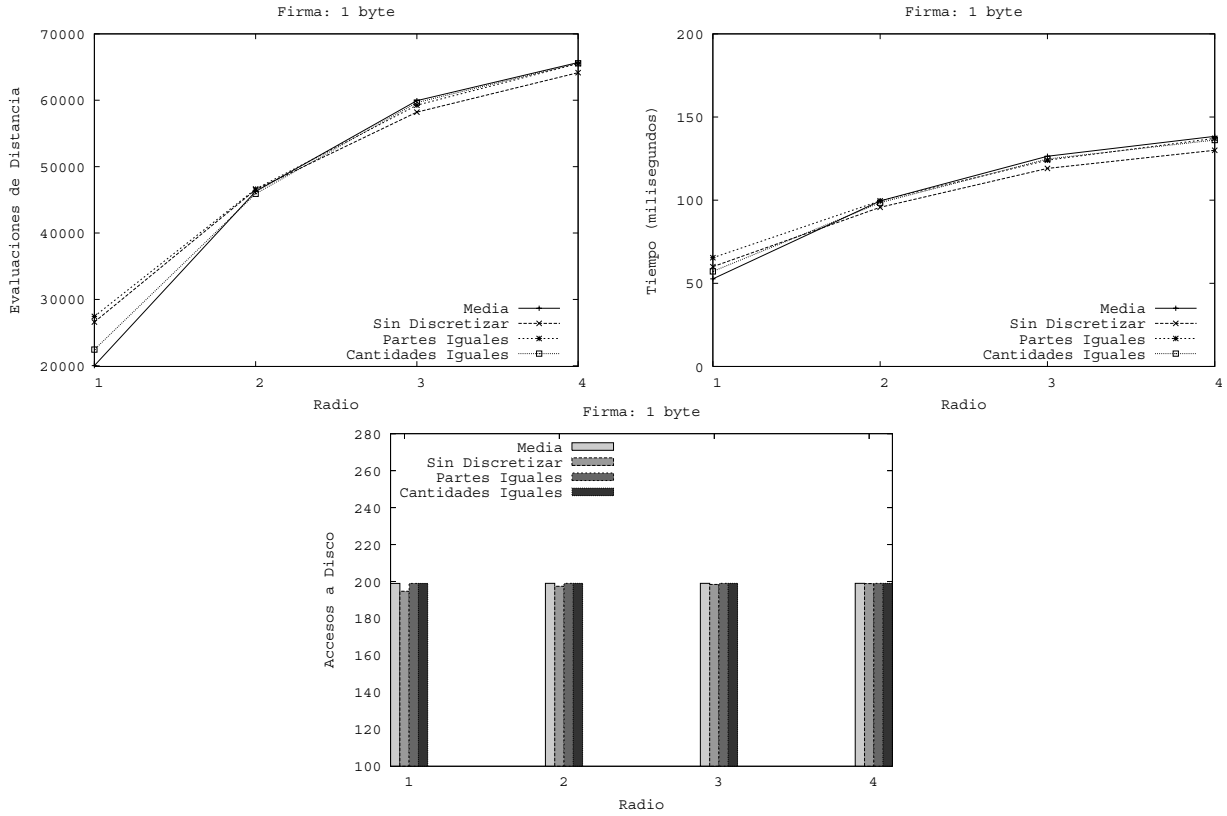


Figura 7: *Diccionario Inglés, tamaño de firma 1 byte*

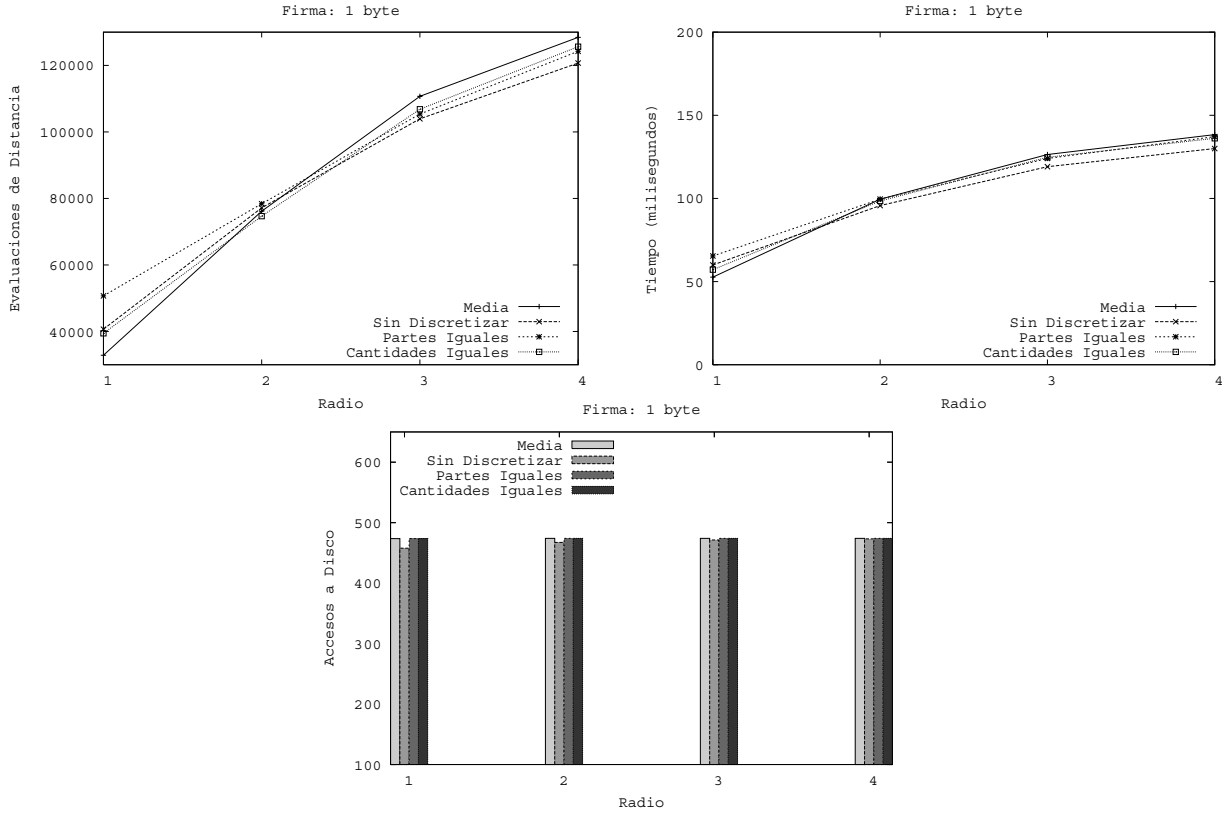


Figura 8: *Diccionario Francés, tamaño de firma 1 byte*

5. Conclusiones y Trabajo Futuro

En este trabajo combinamos el método de paginado PLCS con los métodos de discretización *media, partes iguales y cantidades iguales* con el objetivo de obtener una implementación del FQTrie que sea eficiente en términos de las tres componentes que afectan el tiempo de resolución de una consulta: cantidad de evaluaciones de la función de distancia d, cantidad de accesos a disco y tiempo extra de CPU.

La técnica PLCS con *media* resultó ser la más competitiva para búsquedas de alta selectividad y tamaños de firmas pequeños (1 y 2 bytes). La técnica PLCS sin discretizar debió cuadruplicar el espacio, usando firmas de 4 bytes, para lograr superar a PLCS con *media* en búsquedas de alta selectividad. Si se consideran búsquedas de baja selectividad PLCS sin discretizar resulta ser la opción más conveniente.

En cuanto al trabajo futuro nos proponemos estudiar en detalle las causas por las cuales PLCS con *media* no es competitivo en búsquedas de baja selectividad y solucionar dichos problemas. Además, estamos analizando métodos de paginado que puedan ser usados sobre espacios métricos cuyos objetos ocupan más de una página de disco.

Referencias

[1] E. Chávez and K. Figueroa. Faster proximity searching in metric data. In *Proceedings of MICAI 2004. LNCS 2972, Springer*, Cd. de México, México, 2004.

[2] E. Chávez, J. Marroquín, and G. Navarro. Fixed queries array: A fast and economical data structure for proximity searching. *Multimedia Tools and Applications (MTAP)*, 14(2):113–135, 2001.

[3] E. Chávez, G. Navarro, R. Baeza-Yates, and J.L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, September 2001.

[4] E. Chávez, N. Herrera, C. Ruano, and A. Villegas. Funciones de discretización basadas en histogramas de distancia. In *Actas de la Conferencia Latinoamericana de Informática (CLEI'06)*, Santiago, Chile, 2006.

[5] G. H. Gonnet and R. Baeza-Yates. *Handbook of Algorithms and Data Structures*. Addison-Wesley, 1991.

[6] A. Villegas, E. Chávez, and N. Herrera. Métodos de paginación para índices métricos basados en pivotes. In *Actas del X Congreso Argentino de Ciencias de la Computación (CACIC'04)*, pages 306–316, Buenos Aires, Argentina, 2004.

PICS un Sistema de Comprensión e Inspección de Programas

Mario M. Berón
Universidad Nacional de San Luis - Departamento de Informática
San Luis - Argentina
mberon@unsl.edu.ar
Pedro R. Henriques
Universidad de Minho - Departamento de Informática
Braga - Portugal
prh@di.iminho.pt
Maria J. Varanda
Instituto Politécnico de Bragança - Departamento de Informática
Bragança - Portugal
mjoao@ipb.pt
Roberto Uzal
Universidad Nacional de San Luis - Departamento de Informática
San Luis - Argentina
ruzal@sinectis.com.ar

Resumen

La Comprensión de Programas (CP) es un área de la Ingeniería del Software que tiene como objetivo el estudio y creación de modelos, métodos, técnicas y herramientas con la finalidad de facilitar el entendimiento de los sistemas de software. La CP es útil para mantenimiento, reingeniería, ingeniería reversa, entre otras tantas aplicaciones.
Para construir herramientas de comprensión de programas es necesario concebir esquemas que permitan extraer y visualizar la información de los sistemas. PICS es una herramienta de CP basada en la presentación de diferentes perspectivas o vistas del sistema de estudio. Estas vistas tienen como propósito interrelacionar distintos dominios útiles para la comprensión. En este contexto, la principal fortaleza de PICS es alcanzar una estrecha relación entre el dominio del programa y el comportamiento del sistema. Esta relación falicita la comprensión porque permite asignar semántica a las componentes del dominio del programa.
En este artículo describimos PICS, sus funcionalidades de inspección, comprensión y las estratégias implementadas en su núcleo que permiten alcanzar la interrelación de dominios. En este último caso, hacemos énfasis en la interrelación entre el dominio del programa y el comportamiento del sistema debido a la gran importancia que tiene esta relación para la CP.
Palabras Claves: Comprensión de Programas, Métodos, Técnicas, Herramientas.

1. INTRODUCCIÓN

La comprensión de programas se traduce en la habilidad de entender una pieza de código escrito en un lenguaje de alto nivel. Un programa no es mas que una secuencia de instrucciones que serán ejecutadas de forma de garantir una determinada funcionalidad. El lector de un programa consigue extraer el significado del mismo cuando comprende de que forma el código cumple con la tarea para la cual fue creado. El área de comprensión de programas es una de las más importantes de la Ingeniería del Software porque es necesaria para tareas de reutilización, inspección, manutención, migración y extensión de sistemas de software. Puede también ser utilizada en áreas como ingeniería reversa o enseñanza de lenguajes de programación. La tarea de comprensión de programas puede tener diferentes significados y puede ser vista desde diferentes perspectivas. El usuario puede estar interesado en como la computadora ejecuta las instrucciones con el objetivo de comprender el flujo de control y de datos, o puede querer verificar los efectos que la ejecución tiene sobre el objeto que esta siendo controlado por el programa. Considerando estos niveles de abstracción, una herramienta versátil de inspección visual de código es crucial en la tarea de comprensión de programas [6].
En este artíulo se presenta PICS (**P**rogram **I**nspection and **C**omprehension **S**ystem) una herramienta destinada a facilitar el proceso de comprensión de programas escritos en lenguage C. PICS aborda este desafío por medio de la presentación de diferentes vistas. Estas vistas representan al sistema de estudio en distintos dominios y en diferentes niveles de abstracción. La presentación de vistas por si sola no es suficiente son necesarios mecanismos de navegación entre ellas. Esto posibilita que el programador pueda acceder a un dominio específico y cuando considere necesario acceder a otro con facilidad. Esta característica facilita el aprendizaje debido a que fomenta la interrelación de conceptos. De se esta manera se reduce la brecha existente entre los conocimientos del programador y los conceptos subyacentes del sistema.
Una de las relaciones más importantes es la que vincula el dominio del programa y el comportamiento del sistema. Podemos realizar esta afirmación porque dicha vinculación permite identificar claramente las operaciones realizadas por el sistema y su efecto comportamental. De esta forma la tarea de modificar, documentar o actualizar una funcionalidad es más leve. Esto se debe a la posibilidad acceder solamente a las componentes del sistema utilizadas para producir el comportamiento bajo análisis.
Para alcanzar la interrelación de dominios PICS utiliza técnicas de compilación que permiten extraer información estática y dinámica. Luego construye vistas textuales y basadas en grafos que representan al sistema o parte de él en otros dominios [3]. Además, PICS implementa una estrategia de relación operacional-comportamental denominada BORS (**B**ehavioral-**O**perational **R**elation **S**trategy) que posibilita encontrar la relación entre las componentes del dominio del programa y el comportamiento del sistema.
Este artículo esta organizado de la siguiente manera. La sección 2 describe la arquitectura de PICS. La sección 3 explica las técnicas de compilación utilizadas para la extracción de información estática y dinámica. La sección 4 describe brevemente la información disponible en PICS y la forma de administrarla. La sección 5 detalla las vistas y sus funcionalidades. La sección 6 explica BORS. Finalmente, la sección 7 expone la conclusión de este artículo.

2. ARQUITECTURA DE PICS

La Figura 1 muestra la aquitectura de PICS. El lector puede observar que la herramienta posee las siguientes componentes:

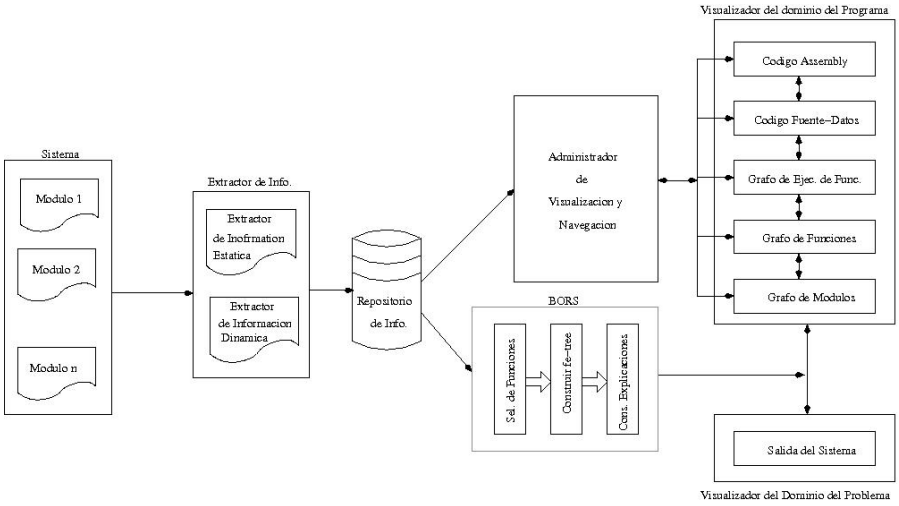


Figura 1: Arquitectura de PICS

Módulos para extracción de la información: Estos módulos extraen información estática y dinámica del sistema de estudio. Para la extracción de la información estática se utilizan técnicas de compilación tradicionales. Para la recuperación de la información dinámica se instrumenta el código fuente con funciones de inspección. Este esquema de instrumentación será descripto brevemente en la sección 3.

Repositorio de Información: Esta componente posibilita el almacenamiento de toda la información extraída. Los módulos principales que acceden a este repositorio son el *Administrador de Visualización y Navegación* y *BORS*. Esta componente será presentada brevemente en la sección 4.

Administrador de Visualización y Navegación: La función principal de esta componente (ver sección 4) es la de posibilitar la construcción de las vistas disponibles en PICS. Además permite la navegación entre ellas.

Visualizadores: Esta parte del sisema consta de varias componentes agrupadas en dos categorías (Visualizaciones del Dominio del Programa y Visualizaciones del Dominio del Problema). Cada componente implementa las distintas vistas disponibles en PICS. Cada una de ellas, además de construir la vista apropiada, usando la información provista por el *Administrador de Visualización y Navegación*, presentan al usuario una visión gráfica o textual dependiendo de la clase de perspectiva que se utilize. Las vistas provistas por PICS serán descriptas en sección 5.

BORS: Este módulo es el encargado de implementar la estrategia de relación operacional-comportamental. En otras palabras, permite relacionar el dominio del programa con el dominio del problema. La estrategia *BORS* es compleja y será descripta en la sección 6.

3. EXTRACCIÓN DE LA INFORMACIÓN

Para la extracción de la información estática PICS utiliza técnicas de compilación tradicionales. Es decir, la herramienta tiene implementado un analizador sintáctico de ANSI-C con

las acciones semánticas necesarias para la recuperación de: funciones, datos, tipos, etc.
Para el caso de la recuperación de la información dinámica [5] [8] [2] se utilizó *instrumentación de código*. Esta técnica consiste en insertar sentencias útiles dentro del código fuente del sistema. Esto tiene por objetivo extraer la información deseada a medida que el sistema se ejecuta. El esquema utilizado para este caso es simple: *se insertan funciones de inspección en el comienzo y fin de las funciones*. Estas funciones de inspección tienen como tarea imprimir el nombre de la función ejecutada y alguna otra información que el usuario desea capturar. Este esquema se debe completar con una estrategia para controlar el número de funciones recuperadas por las funciones de inspección. Esto se debe a que las funciones pueden ser invocadas dentro de iteraciones. Las iteraciones pueden repetirse muchas veces porque normalmente son utilizadas para inicializar estructuras de datos o porque forman parte de un algoritmo complejo. En este caso, las funciones del sistema se invocaran en forma repetitiva y las funciones de inspección registrarán este hecho. Por esta razón, la información recuperada será enorme. Para resolver este problema las iteraciones son controladas insertando código antes dentro y después de ellas. El código preliminar a las iteraciones indica, a las funciones de inspección, el número de veces que las funciones invocadas dentro del cuerpo de la iteración deben ser reportadas. Este número es almacenado en una pila debido a que las iteraciones pueden estar anidadas. El código dentro de las iteraciones tiene como objetivo decrementar el número que está en el tope de la pila (número de veces que las funciones dentro de la iteración pueden mostrarse). Cuando este valor llega a cero las funciones de inspección no recuperan los datos de las funciones invocadas. Finalmente, el código insertado después de las iteraciones tiene como finalidad recuperar el número de veces que las funciones de la iteración anterior debe mostrarse. Dicho valor se encuentra en el tope de la pila de control de ciclos. La Figura 2.a muestra el esquema para las funciones de inspección y la Figura 2.b muestra el esquema descripto para las iteraciones. El lector interesado en una descripción detallada de esta aproximación puede leer [5].
Para finalizar esta sección es importante notar que toda la información (estática y dinámica) es almacenada en el *Repositorio de Información*. Dicha componente es accedida por otros módulos del sistema para la construcción de vistas y otras operaciones.

```
void f (int a, int b)
  {
    INSPECTOR_ENTRADA("f");
    ........
    INSPECTOR_SALIDA("f");
    return;
  }
```

(a)

```
push(pila,N)
for(i=0;i<TAM;i++)
 {
  /* acciones del loop */
  .....
  decrementarTope(pila);
 }
pop(pila);
```

(b)

Figura 2: Instrumentación de Funciones y Control de Ciclos

4. ADMINISTRACIÓN DE LA INFORMACIÓN

Esta componente almacena información de: tipos, módulos, variables, funciones y relaciones existentes entre ellas. Entre las relaciones más destacadas encontramos: llamadas a funciones,

comunicación y dependencia de módulos, dependencia de tipos, definiciones de variables clasificadas por funciones y módulos, etc.
Por otra parte, esta componente administra la información dinámica del sistema. Como por ejemplo las funciones y tipos usados en tiempo de ejecución.
Tanto la información estática como la dinámica son administradas usando un modelo relacional de base de datos. La parte más compleja de este módulo radica en la administración de la información dinámica. Esta complejidad se debe a la enorme cantidad de información recuperada por las técnicas de extracción de información dinámica. Este problema ha conducido al uso de organizaciones de archivos sofisticadas para la implementación de esta componente.

5. VISTAS

Una vista es una perspectiva del sistema de estudio que permite resaltar algún aspecto del mismo. Las vistas son importantes porque actúan como facilitadores del aprendizaje y por lo tanto ayudan en la tarea de CP. En las subsecciones siguientes se describen los objetos de interés para visualizar e inspecionar [4], las vistas y sus funcionalidades.

5.1. OBJETOS VISUALIZADOS POR PICS

Entre los objetos que pueden ser visualizados por PICS se pueden mencionar:

Código Fuente y Objeto: los códigos fuente y objeto son las unidades básicas de inspección. Generar estrategias que faciliten la lectura y navegación en ambos códigos facilita la comprensión e inspección.

Datos: los datos son una fuente importante de información. A través del análisis de los datos el usuario puede:

- Extraer el mapa de tipos (un grafo que muestra los tipos y las relaciones entre ellos).
- Relacionar las funciones con los datos. Esta operación es muy compleja cuando el sistema de estudio esta implementado usando el paradigma imperativo.
- Visualizar la tabla de símbolos del sistema.
- etc.

Grafo de Llamadas a Funciones: permite ver, en forma estática, como las funciones del sistema están interconectadas. Esta perspectiva es tradicional en CP y es muy útil para detectar, mediante operadores de grafos, funciones que son puntos de entrada o salida de un sistema, funciones escenciales etc. También es muy adecuado para usarlo en combinación con técnicas de análisis dinámico.

Funciones de Tiempo de Ejecución: consta de las funciones recuperadas por la técnica de instrumentación de código descripta en la sección 3. Conocer las funciones usadas por el sistema para alcanzar un objetivo simplifica el proceso de inspección y comprensión.

Grafo de Comunicaciones de Módulos: posibilita visualizar, en forma estática, como los módulos del sistema están interconectados. Esta perspectiva, al igual que el grafo de llamadas a funciones, es tradicional en CP. Sobre ella se pueden aplicar operadores de grafo para detectar módulos con características particulares.

Salida del Sistema: útil para entender las funcionalidades del sistema. El mecnanismo de inspección dinámica implementado en PICS posibilita relacionar la salida del sistema con las componentes del programa utilizadas. De esta manera se logra una visión integrada entre el comportamiento y la operación del programa.

5.2. AMBIENTES DE VISUALIZACIÓN Y FUNCIONALIDADES DE PICS

En esta subsección se presentan las interfaces que implementan distintas vistas útiles para la comprensión e inspección de programas. Estas vistas posibilitan visualizar las componentes descriptas en la sección 5.1. Además se explica la forma de navegación y las funcionalidades provistas por cada una de ellas.

5.2.1. Interfaz de preprocesamiento e instrumentación de código

PICS posee un ambiente para visualizar el código fuente, preprocesarlo, aplicar el analizador sintáctico y visualizar el código instrumentado. Esta interfaz puede ser vista en la Figura 3. La interfaz esta compuesta de dos ventanas. En la ventana superior, el usuario puede visualizar

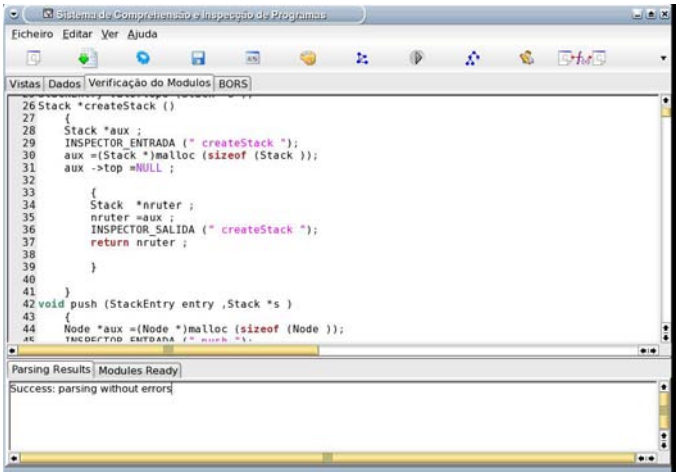


Figura 3: Visualizador de Código Original, Preprocesado o Instrumentado

el código original, preprocesado o instrumentado. En la ventana inferior se pueden ver los resultados de cada una de las operaciones. La visualización del código original puede llevarse a cabo con una simple operación abrir asociada al primer botón de la barra de herramientas. El preprocesamiento se lleva a cabo presionando el segundo botón. El resultado de esta operación se muestra automáticamente en la ventana superior. Finalmente, luego del preprocesamiento, el código se puede instrumentar presionando el tercer botón. En la Figura 3 el lector puede ver el código de un sistema instrumentado con las funciones de inspección.

5.2.2. Visualización de la tabla de símbolos

La Figura 4 muestra la interfaz para visualizar la tabla de símbolos provista por PICS. Esta interfaz consta de dos ventanas. La superior contiene el código fuente bajo análisis. En la

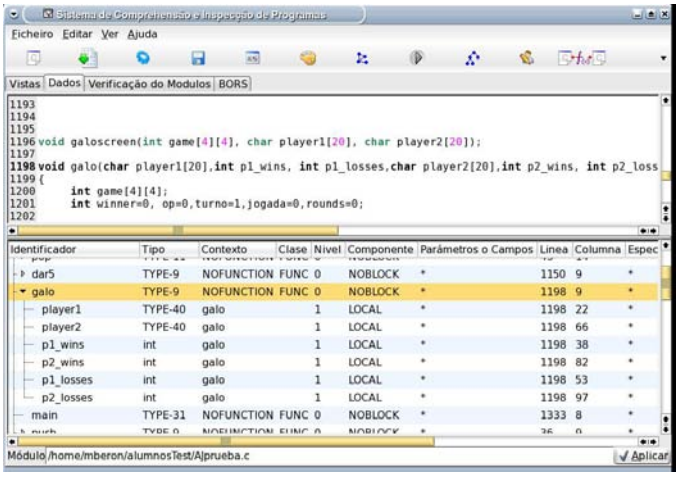


Figura 4: Visualizador de la Tabla de Símbolos

inferior se puede observar cada uno de los identificadores definidos en el sistema. Estos identificadores son clasificados en *Tipos Anónimos*, *Primitivos*, *Variables*, *Funciones* y *Estructuras*. Los primeros describen los tipos anónimos generados por el analizador sintáctico. Los segundos son los tipos predefinidos por el lenguaje usado para implementar el sistema, en este caso C. Los terceros referencian a las variables. Finalmente, los últimos identifican a las funciones del sistema y a sus parámetros como asi también a las estructuras y sus campos.
El mecanismo de navegación tiene dos modos de operación: *automático* y *búsqueda*. En el modo *automático*, el usuario sólo necesita posicionar el puntero del ratón sobre el identificador deseado. El sistema muestra, en forma automática en la ventana superior, el lugar donde aparece ese identificador en el código fuente. El modo *búsqueda* se utiliza cuando el usuario desea buscar un identificador. En este caso es necesario estar posicionado en la ventana inferior y luego escribir las letras que componen al identificador deseado. En ese momento se activa una búsqueda fonética que va seleccionando el identificador que contiene las letras escritas. Luego de realizada esta tarea solamente es necesario presionar la tecla *enter* para que el sistema muestre el identificador en la ventana superior.

5.2.3. Visualización de tipos de datos abstractos

La Figura 5 muestra el visualizador de *Tipos de Datos Abstractos* (TDAS) detectados por una heurística de detección de tipos incorporada en PICS. Este visualizador consta de un comando y dos ventanas principales. El comando esta disponible en la barra de herramientas. Basicamente este comando permite cargar, en la ventana principal izquiera, el archivo que contiene los TDAs del sistema. Esta ventana organiza la información en forma de árbol con tres niveles. La raíz (nivel 0) vincula a todos los nombres de los TDAS (nivel 1). Cada nombre de TDA contiene las funciones asociadas por la heurística de dectección de TDAS (nivel 2).
Cada nodo del árbol de tipos tiene asociado una operación de inserción. Esta operación permite que el usuario pueda colocar funciones de la ventana de la izquierda en la ventana de la derecha. Esta última ventana contiene las funciones que el usuario desea explicar y que sirven como entrada a la estrategia de relación operacional-comportamental denominada BORS. Los elementos incorporados en la ventana principal derecha tienen una función asociada que posibilita la eliminación del mismo. Esto tiene como objetivo permitirle al usuario personalizar

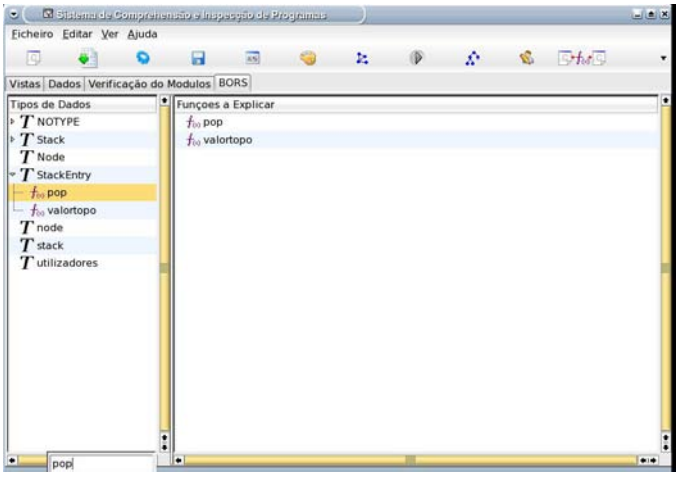


Figura 5: Visualizador de TDAS

las funciones que desea explicar. Finalmente, ambas ventanas estan dotadas de funciones de búsqueda fonética automática para facilitar la interacción del usuario con PICS.

5.2.4. Visualizaciones basadas en grafos

Para la visualización de grafos (de funciones, módulos o tipos) PICS posee un visualizador con las siguientes funcionalidades:

- Visualización del grafo correspondiente.
- Operaciones de zoom y movimiento del grafo.
- Cuando se dispone de información de las funciones usadas en tiempo de ejecución permite realizar animaciones a nivel funciones y módulos. Esas animaciones consisten en resaltar, con un color determidado, las funciones o módulos usados con un intérvalo de tiempo de demora.
- Animación paso a paso usando la información dinámica.
- Inserción y eliminación de nodos y arcos

La Figura 6 muestra el grafo de llamadas a funciones para una aplicación. Como se puede observar el ambiente tiene un sector de visualización donde se dibuja el grafo recuperado. Luego en la parte superior se encuentran diferentes entradas que perminten personalizar el ambiente. Estas entradas posibilitan cambiar de color a los nodos y a los arcos y establecer la demora para el modo animación. En la parte inferior se encuentra un sector de comandos que posibilita la ejecución de la animación o realizar dicha tarea paso a paso. También en la parte inferior, se localiza una entrada que indica el archivo que contiene las funciones de tiempo de ejecución para realizar la animación. Con este ambiente es posible navegar entre el grafo de módulos, el grafo de funciones y el código fuente. Estas operaciones pueden ser realizadas facilmente haciendo click con el ratón en el objeto deseado.

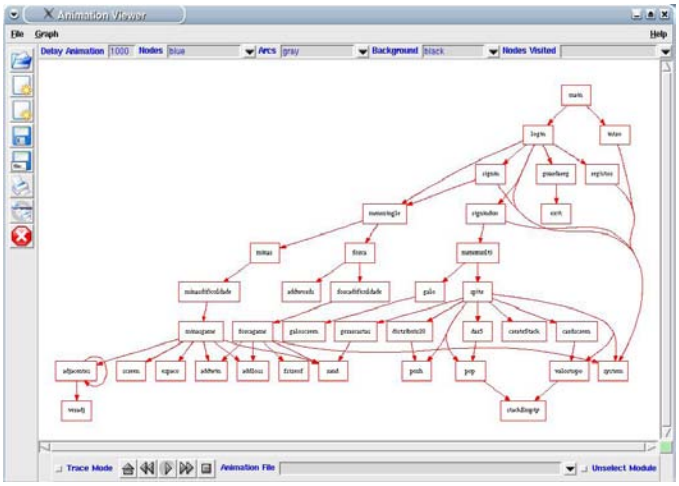


Figura 6: Ambiente de visualización de grafos

5.2.5. Inspección de funciones de tiempo de ejecución

La Figura 7 muestra la interfaz de navegación entre: las funciones usadas en tiempo de ejecución, el código fuente y objeto.
En la ventana de la izquierda el usuario puede observar las funciones de tiempo de ejecución.

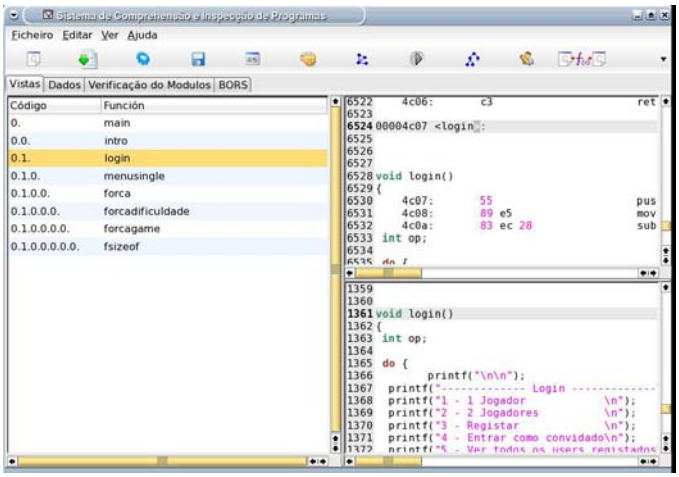


Figura 7: Interfaz de Navegación entre funciones de tiempo ejecucion y código fuente y objeto

En las otras dos se pueden ver el código objeto (a la derecha en la parte superior) y el código fuente (a la derecha en la parte inferior). La navegación se produce haciendo click en la función de tiempo de ejecución deseada. El sistema, usando el repositorio de información, detecta el módulo fuente y objeto correspondiente. Luego de esa tarea, desensambla el módulo objeto para obtener el código assembly y bytecode correspondiente. Finalmente, los archivos resultantes son visualizados, resaltando la función correspondiente, en las ventanas de interfaz de PICS.

5.2.6. Visualización de la relación entre el dominio del programa y el dominio del problema

Con PICS es posible construir una vista que relaciona la operación del programa con su resultado. La Figura 8 muestra esta vista. El lector puede observar tres ventanas. En la

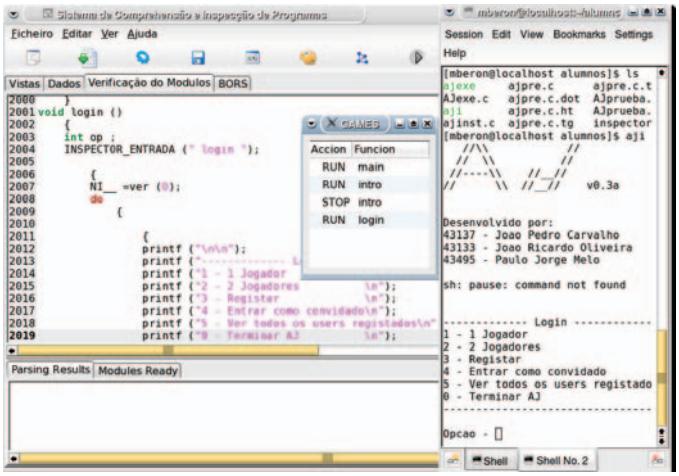


Figura 8: Relación Operacional-Comportamental

ventana del fondo se encuentra la interfaz de visualización de código. También es posible ver: la ventana de funciones ejecutadas (recuperadas por el esquema de instrumentación de código) y la que contiene al sistema ejecutando. El lector puede observar la relación directa entre las funciones (dominio del programa) y el funcionamiento del sistema (dominio del problema). De esta manera el usuario de PICS puede encontrar *significado y sentido* a las operaciones del programa. También esta visualización brinda la información necesaria para comenzar a inspeccionar el código en las partes de interés. La inspección se lleva a cabo utilizando las interfaces de PICS descriptas en las secciones anteriores.

6. BORS

La estrategia de relación operacional-comportamental, denifinida por nuestro grupo de investigación, utiliza infromación dinámica y estática del sistema de estudio. Además se basa en la siguiente observación:

"La salida de un sistema esta compuesta de objetos del dominio del problema. Usualmente estos objetos son implementados por tipos de datos abstractos, en el caso de lenguajes imperativos, o por clases, en el caso de lenguajes orientados a objetos. Tanto los TDAs como las Clases tienen objetos de dato que almacenan su estado y un conjunto de operaciones que los manipulan. Entonces es posible describir cada objeto del dominio del problema utilizando los TDAs o clases que los implementan."

Esta estrategia denominada BORS (Behavioral-Operational Relation Strategy) aplica los siguientes pasos para alcanzar su objetivo.

Detectar las funciones relacionadas con cada objeto del dominio del problema: esta tarea es llevada a cabo en forma semi-automática. El usuario selecciona los TDAs que

desea explicar utilizando la interfaz de TDAs. Entonces todas las funciones relacionadas con esos tipos son cadidatas a ser explicadas y son almacenadas en una lista.

Construir un árbol de ejecución de funciones usadas en tiempo de ejecución: la salida del esquema de instrumentación de código contiene la información suficiente (inicio y fin de la ejecución de las funciones) para construir una estructura de datos denominada fe-tree [1] [7](**f**unction **e**xecution **t**ree) que describe como las funciones del sistema de estudio son o fueron ejecutadas.

Explicar las funciones encontradas en el paso 1 usando el árbol construido en el paso 2: combinando la información de los dos pasos anteriores (lista de funciones a explicar y fe-tree) se pueden explicar las funciones mostrando: el contexto donde fueron invocadas (camino desde la raíz hasta la función) y que tarea realizo para el sistema (sub-árbol cuya raíz es la función que se está explicando). Este proceso se lleva a cabo aplicando un recorrido por niveles sobre el fe-tree. Cada vez que se encuentra un nodo del fe-tree que pertenece a la lista de funciones a explicar se reporta el camino desde ese nodo a la raíz. También es posible mostar el subarbol que tiene como raíz el nodo analizado.

La Figura 9 muestra el procedimiento empleado por esta estrategia. El lector interesado puede

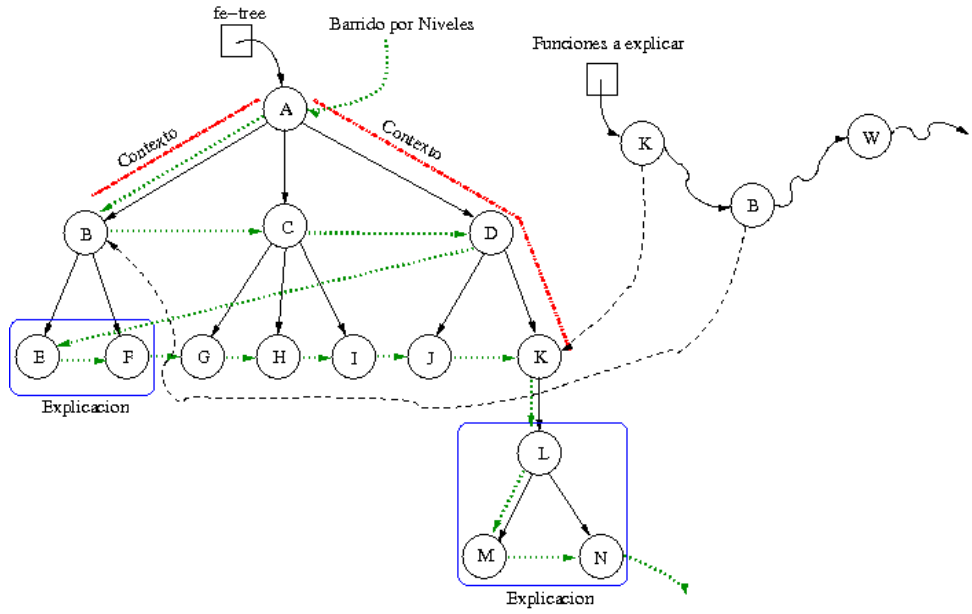


Figura 9: Estructuras de datos, información y procedimiento utilizado por BORS

encontrar en [7] una explicación detallada de la estrategia BORS.

7. CONCLUSIÓN

En este artículo se presentó PICS una herramienta para la inspección y comprensión de programas escritos en lenguaje C. Una de las mayores fortalezas de este sistema consiste en la presentación de diferentes vistas útiles del sistema de estudio. Destacándose, en ese contexto, la relación operacional y comportamental obtenida a través de la implementación de la instrumentación de código y de la estrategia BORS. Esta relación puede ser claramente visualizada

debido a la posibilidad de ejecución en paralelo del sistema bajo estudio y del sistema de inspeción de código.
La estrategia BORS combina información estática y dinámica con el fin de explicar los objetos del dominio del problema usando elementos del dominio del programa. Para llevar a cabo esta tarea es necesario que el usuario indique que tipos o funciones desea explicar. El resto de las actividades se realizan en forma automática.
PICS es el resultado de muchos esfuerzos de investigación e implementación en el contexto de: *modelos cognitivos*, *visualización de software* y *procesamiento de lenguajes*. PICS ha sido aplicado con éxito a diferentes programas cuyos tamaños varian entre las 0.3 kloc y 5 kloc de código C. El resultado de algunas de esas experiencias pueden ser vistos en [7] [5]. Sin embargo, en los últimos tiempos se realizaron otras pruebas interesantes del sistema. Por razones de extensión, y por desear presentar en forma completa el sistema de inspección y comprensión, no fue posible describir estos experimentos en este artículo.
Como trabajo futuro se proyecta incorporar técnicas de generación de documentación de alto nivel desarrolladas por el grupo de investigación PCViA (**P**rogram **C**omprehension by **Vi**sual **I**nspection and **A**nimation) de la Universidad de Minho. Además se desea profundizar en técnicas de extracción de la información y visualización de programas. El principal objetivo es producir visualizaciones de alto nivel para lograr otras formas de relacionar el dominio del programa y del problema.

Referencias

[1] Abdelwahab Hamou-Lhadj.*The Concept of Trace Summarization.*PCODA: Program Comprehension through Dynamic Analysis.(2005).

[2] Andy Zaidman, Bram Adams, and Kris Schutter. *Applying Dynamic Analysis in a Legacy Context: An Industrial Experience*. PCODA: Program Comprehension through Dynamic Analysis (2005).

[3] Franoise Balmas, Harald Werts, Rim Chaabane. *DDGraph: a Tool to Visualize Dynamic Dependences.*Program Comprehension through Dynamic Analysis (2005).

[4] Maria J. Pereira. *Sistematização da Animação de Programas*. Ph.D. thesis, Universidade do Minho, Braga, 1996.

[5] Mario M. Berón, Pedro Henriques, Maria J. Varanda, Roberto Uzal, Germán Montejano. *Language Processing Tool for Program Comprehension*. XII Argentine Congress on Computer Science(2006).

[6] Mario M. Berón, Pedro Henriques, Maria J. Varanda, Roberto Uzal. *Herramientas para la comprensión de programas*. VIII Workshop de Investigadores en Ciencias de la Computación (2006).

[7] Mario M. Berón, Pedro R. Henriques, Maria J. Varanda Pereira, Roberto Uzal. *Static and Dynamic Strategies to Understand C Programs by Code Annotation*. European Join Conference on Theory and Practice of Software. Braga-Portugal. 2007.

[8] Wang Yuying, Li Qingshan, Chen Ping, Ren Chunde.*Dynamic Fan-in and Fan-out Metrics for Program Comprehension*. PCODA: Program Comprehension through Dynamic Analysis (2005).

PROCESO DE AGREGACIÓN CON MÚLTIPLES NIVELES DE EVIDENCIA PARA ESTUDIOS EXPERIMENTALES EN INGENIERÍA DEL SOFTWARE

Enrique Fernández[1,2], Paola Britos[1,2], Ramón García-Martínez[1,2], Oscar Dieste[3]
[1]Centro de Ingeniería de Software e Ingeniería del Conocimiento. Escuela de Postgrado. ITBA. Argentina
[2]Laboratorio de Sistemas Inteligentes. Facultad de Ingeniería. Universidad de Buenos Aires. Argentina
[3]Grupo de Investigación en Experimentación de Software. Facultad de Informática. Universidad Politécnica de Madrid.
España
rgm@itba.edu.ar

Abstract

Current process' of aggregation of result about experimental studies, based on meta analysis from other sciences, showed that they are not suitable for software engineering. This article adds an alternative aggregation process to the standard one, based on an aggregation strategy with multiple levels of evidence; in which each level is connected with a specific aggregation technique that is assigned considering the quality and quantity of experimental studies identified.

Keywords: Aggregation with multiple levels of evidence, meta analysis

Resumen

Los actuales procedimientos de agregación de resultados de estudios experimentales basados en Meta-Análisis, provenientes de otras ramas de la ciencia, han mostrado no ser adecuados para la realidad que vive la Ingeniería del Software. El presente trabajo presenta un proceso de agregación alternativo al estándar basado en una estrategia de agregación con múltiples niveles de evidencia. Donde cada uno de los niveles de evidencia se encuentra vinculado con una técnica de agregación específica que se asigna en función de la calidad y cantidad de estudios experimentales identificados.

Palabras claves: Agregación con múltiples niveles de evidencia, meta-análisis

1. Introducción

Desde hace varios años, la cantidad de experimentos realizados dentro del ámbito de la Ingeniería del Software (SE) se ha incrementado significativamente. Estos experimentos abarcan los más variados temas, como ser: el desempeño de las técnicas de prueba, la educción de requisitos, o el desempeño de los lenguajes de programación, entre otros. Si bien los experimentos aportan conocimientos interesantes en cada caso, para que la información que aportan sea valiosa los resultados deben agregarse para poder obtener conclusiones avaladas con el mayor número de estudios posibles.

Ha habido algunos intentos de síntesis de experimentos para IS, por ejemplo [1], [2], [3], [4], [5], [6]. Pero todos estos esfuerzos no han dado los resultados esperados. También los resultados de las combinaciones informales [2], [4], [5], [6] fueron limitados. Por otra parte los intentos de combinaciones con técnicas estadísticas [1], [3] resultan impracticables debido a las diferencias en el diseño y ejecución de los experimentos realizados.

Recientemente, se ha propuesto la Revisión Sistemática (RS) [7], [8] como un método para sistematizar la agregación de estudios empíricos en IS.

Una Revisión Sistemática es un procedimiento que aplica estrategias científicas para aumentar la fiabilidad del proceso de recopilación, valoración crítica y agregación de los estudios experimentales relevantes sobre un tema [9]. Las RS se han comenzado a emplear recientemente en IS [10], [11]. No obstante, si bien las RS proporcionan un marco de trabajo que permite realizar la recopilación de experimentos y, en menor medida, la valoración crítica de los mismos, esta falla a la hora de agregar los resultados. El motivo de la falla reside en que la estrategia de agregación utilizada por las RS es Meta-Análisis. El Meta-Análisis es un nombre colectivo que hace referencia a un conjunto de métodos estadísticos que permiten combinar resultados experimentales, siempre y cuando se verifiquen ciertas restricciones, tales como un número mínimo de experimentos, adecuadamente recopilados y homogéneos [12]. Estas restricciones distan de cumplirse en IS. Así por ejemplo: en [11] no puede aplicarse Meta-Análisis debido a la heterogeneidad entre los experimentos y en el caso del problema fue la carencia de replicaciones.

Por consiguiente es necesario desarrollar un procedimiento de agregación específico que considere las características particulares de un entorno de investigación poco maduro como es la IS, de modo que sea posible la aplicación de dicho procedimiento: con un bajo número de replicaciones y con heterogeneidad de estudios, entre otras características. Este procedimiento permitirá mejorar el nivel de las conclusiones obtenidas sobre el desempeño de técnicas y herramientas que a diario se utilizan en IS, muchas veces adoptadas o adquiridas por la fama de quienes las patrocinan. Dando así a la IS un entorno de desarrollo propio de la Ingeniería.

La sección 2 de este artículo describe el estado del arte; la sección 3 describe la problemática identificada en esta etapa de la investigación; la sección 4 define la importancia de problema tratado; la sección 5 establece los materiales y métodos a ser utilizados para el logro de los objetivos; la sección 6 presenta una introducción a la primer versión de la propuesta de solución; la sección 7 describe cual será la estrategia de validación de los resultados. Finalmente, la sección 8 describe algunas de las conclusiones obtenidas hasta el momento.

2. Estado de la Cuestión

La agregación de estudios experimentales tiene una larga historia en disciplinas como la educación o la psicología [13], aunque últimamente su desarrollo está impulsado por las ciencias de la salud [14].

Para agregar los resultados de estudios experimentales, existen dos tipos de métodos: los interpretativos y los no interpretativos [15]. Los métodos interpretativos, tales como "Narrative Summary" [16] o "Grounded Theory" [17], se caracterizan por que las conclusiones son generadas en base a los criterios personales de quienes analizan los resultados [18]. Como estos métodos arrojan resultados poco fiables por su alta dependencia respecto del revisor, su uso ha venido decayendo a favor de los más fiables métodos no interpretativos.

Dentro de los métodos no interpretativos existen varias alternativas, tales como: "Case Surveys" [19], "Vote Counting" [12] o "Análisis Comparativo" [20]. No obstante el más sofisticado de todos los métodos es el Meta-Análisis [21]. Según [22] el Meta-Análisis es el análisis estadístico de una serie de estudios individuales, con el objeto de integrar los resultados en una medida resumen.

Al realizar un Meta-Análisis, deseamos hallar un resultado numérico que sea resumen representativo de los resultados de los estudios individuales, y por tanto que signifique una mejora sobre las estimaciones individuales. En la actualidad el meta-análisis se implementa mediante la estimación del Tamaño de Efecto a través de las Diferencias Medias Ponderadas [12], la cual es conceptualmente sencilla: el estimador de efecto global se calcula como una media ponderada de los estimadores de efecto de los estudios individuales.

Para que el Meta-Análisis arroje resultados representativos de los estudios incluidos, es necesario validar que los estudios individuales puedan resumirse y puedan combinarse. Esta propiedad se conoce con el nombre de "homogeneidad" y se determina a través de la "heterogeneidad estadística", la cual permite identificar si la variación en los resultados de los estudios, es debido a un error aleatorio o no.

Otro aspecto a verificar cuando se desarrolla un Meta-Análisis, es determinar cuan dependiente de los estudios mas robustos es el resultado obtenido, lo que puede estimarse a través de un "análisis de sensibilidad". Este análisis permite ver si todos los estudios fueron tenidos en cuenta en el resultado final obtenido o no.

Diversos autores [7], [8] propugnan la utilización del Meta-Análisis en IS. Sin embargo, como se mencionó mas arriba, hoy día es prácticamente imposible aplicar una estrategia de agregación mediante Meta-Análisis dentro del ámbito de la IS. Los principales motivos que impiden su aplicación son:

- Escasez de experimentos, replicaciones y homogeneidad entre los mismos. [7], [10].
- Carencia de estándares para reportes de experimentos. Por ejemplo, [23] no publican varianzas y [24] ni siquiera reporta las medias de los resultados experimentales. En estas circunstancias la aplicación del Meta-Análisis es imposible.
- Calidad interna variada. Por ejemplo, tratando un mismo tema de investigación, los trabajos de [24] y [25] muestran una gran discrepancia en la concepción y armado del estudio, esto hace que los mismos no pueden considerarse replicaciones y por tanto no puedan utilizarse para un proceso de agregación por Meta-Análisis, es mas, si se hiciese el resultado sería invalidado por el análisis de heterogeneidad.
- Falta de estandarización de las variables respuesta. Por ejemplo, los trabajos de [26] y [27] utilizan diferentes variables respuesta para analizar un mismo aspecto, lo cual hace que estos experimentos no puedan ser considerados replicaciones.

Además del Meta-Análisis existen otras técnicas alternativas menos sofisticadas y que poseen menos restricciones para su aplicación. Dentro de este grupo podemos encontrar: "Case Surveys" [19], "Vote-Counting" [12] y "Análisis Comparativo" [20]. A diferencia del Meta-Análisis, el cual

ha sido estudiado en profundidad, no se han determinado los límites de aplicabilidad de dichas técnicas, con la única excepción del "Vote Counting" [12], y su aplicación en la IS ha sido testimonial hasta la fecha [11], [28]. Por consiguiente, estas técnicas, aunque prometedoras, deben estudiarse extensivamente antes de ser utilizadas rutinariamente en la IS.

3. Descripción del Problema

Actualmente, no existe un método de agregación específicamente adaptado a las necesidades de la IS. Por su parte, los métodos más fiables, como el Meta-Análisis, poseen restricciones que limitan su aplicabilidad. Además, existe un conjunto de métodos que podrían ser aplicables, tales como "Case Survey" o "Análisis Comparativo", pero estos no han sido nunca utilizados en IS y, por lo tanto, se desconoce su idoneidad para realizar agregaciones. En consecuencia, el presente trabajo tiene como objetivo desarrollar un método de agregación que permita obtener la mayor cantidad de piezas de conocimiento posibles, combinando la mayor cantidad de estudios, independientemente de su calidad. El mismo deberá poder trabajar dentro de las limitaciones propias de la IS, esto es: poca cantidad de estudios que traten la misma variable respuesta o tratamiento; falta de estandarización de las variables respuestas; y el reporte de pocas variables estadísticas.

4. Importancia del Problema

Desarrollar experimentos en diferentes lugares (laboratorios e industrias) genera resultados parciales respecto de las capacidades y condiciones de aplicación de las técnicas. Si por ejemplo todos los estudios realizados mostraran que la técnica "a" es mejor que la técnica "b" el proceso de agregación sería muy simple, pero en la realidad esto no es muy habitual, en general los resultados son dispares, con experimentos que muestran a la técnica "a" mejor que la técnica "b" y otros que muestran lo contrario.
El entorno en cual se desarrollaron los experimentos hace que la fiabilidad de los resultados obtenidos varíen de un estudio a otro, estos aspectos deben tenerse en cuenta a la hora de combinar los resultados para que la conclusión sea lo mas fiable posible. Por lo cual para obtener piezas de conocimiento fiables, se debería contar con una estrategia de agregación adecuada a las actuales características de la IS.

5. Materiales y método

Para producir el conocimiento necesarios seguiremos un enfoque clásico [29], [30], [31] identificando los materiales y métodos necesarios para desarrollar nuestro experimento:

Materiales:
Para el desarrollo del método de agregación utilizaremos tres tipos de materiales:

a) Técnicas de agregación en SE: En la actualidad se utilizan en IS técnicas de Meta-Análisis [7], [8], así como técnicas mas informales [4], [5], [10], estas técnicas serán revisadas y, probablemente, utilizadas durante la presente investigación.

b) Técnicas de Agregación en otras disciplinas: Además de las técnicas de agregación utilizadas actualmente dentro del ámbito de la IS, se estudiaran otras técnicas no utilizadas hasta el momento dentro de esta disciplina, como son el "Response Ratio" [31] o "Case Surveys" [19] por citar dos ejemplos.

c) Estudios Experimentales: En la actualidad se cuenta con una gran cantidad de estudios experimentales de las más variadas características. Los mismos se utilizaran para definir las técnicas de agregación (estas deberán poder paliar las falencias propias de los estudios) y validar la factibilidad de uso de las mismas.

Método:
Las tareas a realizar para desarrollar el presente proceso de agregación serán las siguientes:

a) Se buscará identificar técnicas de agregación alternativas al "Effect Size" a través de revisión bibliográfica y consulta a expertos.
b) Se analizará las condiciones de aplicación de las técnicas de agregación encontradas, es decir, bajo que supuestos la técnica es aplicable. Por ejemplo: si la cantidad de sujetos experimentales reportados en cada estudio debe ser el mismo o puede variar de un estudio a otro.
c) Se analizará la fiabilidad de la respuesta estimada por las distintas técnicas de agregación encontradas, es decir, cual es el nivel de error esperado.
d) Se propondrá un método para la aplicación de las técnicas de agregación en función de su fiabilidad y restricciones de aplicación
e) Se propondrá una estrategia para interpretar los resultados obtenidos por las distintas técnicas, la misma estará vinculada con la fiabilidad de la respuesta brindada por la técnica de agregación.

6. Solución Propuesta

Para solucionar el problema de agregación de estudios, se propone una estrategia de agregación múltinivel, en la cual, las técnicas de agregación no se utilicen de forma alternativa y excluyente entre si, sino, que se utilicen de forma complementaria. Así, en función de la cantidad y calidad de estudios encontrados, se aplicarán las técnicas más convenientes y los resultados se analizarán de manera conjunta. Por ejemplo, si se contase con diez experimentos, cuatro agregables por Meta-Análisis y seis que no, para los experimentos no agregables por Meta-Análisis se utilizará alguna técnica de agregación alternativa, y luego los resultados se mostrarán de forma conjunta con los resultados obtenidos por Meta-Análisis. De esta forma se obtendrá más de un nivel de evidencia, el primero vinculado al Meta-Análisis y los siguientes vinculados a las técnicas alternativas que serán menos fiables.
La estrategia de agregación propuesta se estructura en 5 pasos, tal como puede observarse en la figura 1.

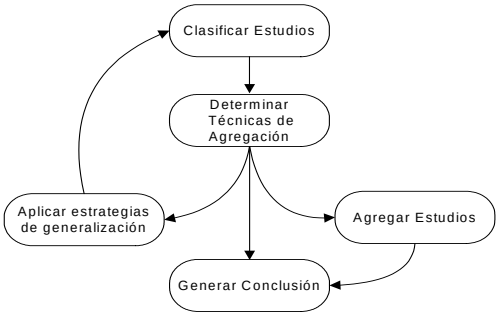


Figura 1. Ciclo de Vida del proceso de Agregación

Clasificar Estudios: La clasificación de estudios tiene como objetivo poder agrupar los distintos estudios en función de su calidad, las variables respuestas que publiquen y los tipos de tratamientos analizados.

Determinar las Técnicas de Agregación: Este paso tiene como objetivo identificar que técnicas de agregación conviene aplicar en función de la cantidad y calidad de estudios encontrados. En caso de no poder aplicar ninguna técnica de agregación, se podrá recomendar realizar algún paso alternativo: "Aplicar Estrategias de Generalización" o "Generar conclusiones".
Aplicar Estrategias de Generalización: La estrategia de Generalización tiene como objetivo paliar los problemas vinculados al bajo número de replicaciones. Para ello se debe buscar características comunes, entre dichos estudios, que permitan agrupar tratamientos y/o variables respuesta dentro de un grupo de mayor nivel de abstracción (más general). Si bien estos grupos de mayor nivel de abstracción no son verdaderas replicaciones, su similitud permite considerarlos como replicaciones conceptuales y, por los tanto, como estudios agregables.
Agregar Estudios: En este paso se aplicarán las distintas técnicas de agregación que permiten combinar los resultados de los estudios experimentales. Esto se hace en base a los criterios y recomendaciones hechos en el paso "Determinar Técnica de Agregación".
Generar Conclusión: El objetivo de esta paso es generar un informe (lo mas fiable posible) con estas piezas de conocimiento. Donde el análisis de resultados comenzará desde los más fiables (obtenidos mediante Meta-Análisis) a los menos fiables (obtenidos mediante técnicas alternativas). De esta forma si los resultados son compatibles (todos los niveles de evidencia afirma que un tratamiento es mejor que el otro) se habrá obtenido una conclusión mas robusta que la que se obtendría aplicando la técnicas en forma aislada. Pero si los resultados no son compatibles, se deberá intentar determinar la existencia de variables aleatorias no identificadas hasta el momento o plantear la necesidad de generar de nuevos experimentos vinculados al tema.

Las siguientes subsecciones describen en detalle cada uno de los pasos antes mencionados:

6.1. Clasificar Estudios

En esta investigación, se van a utilizar un conjunto de técnicas estadísticas que, para asegurar la fiabilidad de la respuesta que estiman, requieren, entre otras cosas, que los estudios empíricos incluidos cumplan con un conjunto de precondiciones [11]. Dichas precondiciones son las siguientes:

- Calidad del Estudio Empírico: grado que define en que medida el estudio ha sido bien diseñado, ejecutado y analizado. Esto permite estimar cuan fiables son los resultados expresados en el estudio. Cuando existen riesgos de que un estudio sea poco fiable, es necesario decidir si debe formar parte de la agregación, y en caso afirmativo, determinar que técnicas de agregación se van a utilizar. Si bien la determinación de la calidad de estudios empíricos es un tema resuelto en la mayoría de las ciencias, esto no esta totalmente definido dentro del ámbito de la IS, por ello para esta primer versión del proceso de agregación se utilizaran las recomendaciones hechas en [8], pretendiendo aplicar en futuras versiones las conclusiones alcanzadas en el proyecto de investigación que se esta llevando a cabo por [33].
- Completitud del Reporte: Este es un segundo aspecto a ser analizado, ya que por bien construido que esté el estudio, si el reporte no refleja un conjunto mínimo de parámetros, las técnicas de agregación no podrán ser aplicadas. Los parámetros más relevantes son: Medias (M), Varianzas (V) y cantidad de sujetos experimentales (N). En caso de no publicarse las "varianzas" se debe identificar si se han publicado los estadísticos: parámetro T de Student o F de Snedecor. Asimismo, en caso de producirse la ausencia de las medias, puede ser un paliativo saber si existió o no diferencias entre las mismas.
- Representatividad de los Tratamientos y Variables Respuesta: como en la actualidad existen pocas replicaciones de estudios, el presente proceso de agregación propone la aplicación de una estrategia de generalización (ver paso "Aplicar Estrategia de Generalización") que permita

incluir dentro de un mismo grupo a tratamientos que no son iguales, pero sus similitudes son mayores que sus diferencias. No obstante, las diferencias entre estos estudios generalizados hacen no aplicable la estimación por "Diferencias Medias Ponderadas" o "Response Ratio". En suma: la generalización de tratamientos limita el tipo de técnica de agregación a utilizar. Lo mismo ocurre con la Representatividad de las Variables Respuesta.

- En base al análisis de los aspectos antes mencionados, se deberá asignar una categoría a cada estudio empírico, esta categoría determina que técnicas de agregación pueden aplicarse al mismo. En la tabla 1, se describen las características principales de los estudios incluidos en cada categoría, así como la técnica que, por el momento, consideramos aplicable a la misma categoría.

La asignación de estudios a categorías puede hacer de modo completamente determinista utilizando la tabla 2. Esta tabla es una primera aproximación a la clasificación de estudios empíricos, y previsiblemente podrá ser actualizada. Sin ir mas lejos, a la tabla 2 se le deberán agregar los resultados sobre calidad de estudios que actualmente investiga [31].
Una vez categorizados los estudios, es necesario hacer una descomposición adicional en función de las variables respuesta que los estudios utilizan. Por ejemplo: Supongamos que se desea determinar cual de dos técnicas de educción llamadas "A" y "B" es mejor. Los estudios empíricos que ensayan dichas técnicas pueden utilizar diversas variables respuestas como el tiempo medio por sesión y la cantidad de conocimiento adquirido. Como estas variables no son compatibles entre si, debemos descomponer el conjunto de estudios disponibles en: "Técnica A vs. Técnica B utilizando la variable respuesta Tiempo medio de sesión" y "Técnica A vs. Técnica B utilizando la variables respuesta cantidad de información". A esta descomposición la llamaremos par Tratamiento-Variable, y guiarán el proceso de contabilización.

Categoría	Características de los Estudios	Técnica de agregación
1	Dentro de esta categoría se aceptan estudios que no posean sesgos y sean similares en cuanto a su confección y dominio de aplicación.	"Diferencias Medias Ponderadas"
2	Dentro de esta categoría se aceptan estudios que sean similares en cuanto a su confección y dominio de aplicación, pero que en lugar de publicar las varianzas, estimen si la diferencia entre medias es significativa o no y publiquen los estadísticos t o F.	"Diferencias Medias Ponderadas" mediante formulas alternativas
3	Dentro de esta categoría se aceptan estudios con defectos leves de reporte (no publican varianzas ni análisis de diferencia entre medias).	"Vote Counting"
4	Dentro de esta categoría se aceptan estudios con defectos graves de reporte (solo expresan diferencias entre medias o dicen que un tratamiento es mejor que otro).	"Conteo de Votos Directo"

Tabla 1. Descripción de la categoría de estudios

Condiciones	R1	R2	R3	R4
Calidad del Estudio Experimental	1-2-3	1-2-3	1-2-3	1-2-3-4-5
El Reporte publica	Medias, Varianzas y Sujetos	Medias, Sujetos y Estadísticos alternativos (t o	Medias y Sujetos	que tratamiento se desempeño mejor

Condiciones	**R1**	**R2**	**R3**	**R4**
		F)		
Tratamientos y Variables Respuesta	Ninguno fue generaliza-do	Ninguno fue generaliza-do	Ninguno fue generaliza-do	Uno o ambos fueron Generalizados
Acciones				
Asignar Categoría	1	2	3	4

Tabla 2. Tabla de decisión para determinar la categoría de los estudios

Una vez establecidos todos los pares Tratamiento-Variable, se deberá determinar la cantidad de estudios asignable por cada par.

6.2. Determinar Técnica de Agregación

Como se mencionó anteriormente, la presente tesis propone utilizar diversas técnicas de agregación de forma conjunta. Para ello, se debe analizar la cantidad de estudios vinculados a cada par Tratamiento-Variable. Esto se debe a que la precisión de las técnicas varía en función de la cantidad de estudios disponibles [34].
Si la cantidad de estudios de categoría 1 es superior a 10 es posible aplicar "Diferencias Medias Ponderadas" [12] para estimar el "Tamaño de Efecto" y dar por terminado el proceso ya que la fiabilidad de la respuesta será muy alta [344]. En caso contrario, si existen estudios de categoría 2, se deberán acumular los estudios de categoría 1 con los de categoría 2 y se recomendará agregarlos por las técnicas de estimación de efectos alternativas [35].
En este caso parece ser suficientes unos 10 artículos para dar por finalizado el proceso. En caso contrario, como las técnicas que vamos a utilizar a partir de ahora son menos precisas, siempre que se puedan aplicar, se recomendará su uso. Cuando se analicen los estudios de categoría 3, se verificará que existen estudios para esta categoría y, en caso de existir, se los acumulará con los de categoría 1 y 2 y se recomendará aplicar la técnica de "Vote Counting" [12]. Por último, si existen estudios de categoría 4, se acumularán los estudios de las cuatro categorías y se recomendará aplicar la técnica "Conteo de Votos Directo".
Las técnicas indicadas son las estudiadas hasta el momento, no obstante, no se descarta la utilización de otras técnicas de agregación a medida que la investigación avance.

Categoría	**Técnicas de Agregación**
1	▪ "Diferencias Medias Ponderadas" [12]
2	▪ "Alternatives" [34]
3	▪ "Vote Counting" [12]
4	▪ "Conteo de Votos Directo"

Tabla 3. Relación entre las categorías y las técnicas de Agregación

6.3. Aplicar Estrategias de Generalización

La generalización tiene como objetivo poner de manifiesto los aspectos comunes, de mayor nivel de abstracción de dos tratamientos o variables respuesta. Con este proceso intentamos salvar los problemas de baja cantidad de replicaciones y falta de estandarización de variables respuesta que, habitualmente, sufren los experimentos hechos en IS. Un ejemplo de generalización: Supongamos que intentando determinar si el lenguaje "C++" es mejor o no que su antecesor el lenguaje "C" se contara solo con 2 estudios que comparen estos dos lenguajes en forma directa. Esta evidencia sería muy baja para llevar adelante un proceso de agregación que genere resultado fiables. Pero, si además de estos dos estudios, se contara con otros estudios que comparen "Delphi" y "Pascal", como "C++" y "Delphi" son Orientados a Objetos y el "C" y el "Pascal" son estructurados, podemos conjeturar que "juntando" o generalizando "C++" y "Delphi" (así como "C" y "Pascal") podremos obtener una conclusión con mayor fiabilidad, debido al mayor número de estudios disponibles. Obviamente, estas conclusiones no responden a la pregunta "es mejor C++ que C", pero si a otra muy similar que genera conocimiento sobre la primera.

6.4. Agregar Estudios

En este paso se aplicarán las distintas técnicas de agregación que permiten combinar los resultados de los estudios experimentales. Esto se hará en base a los criterios y recomendaciones hechas en el paso Establecer Técnica de Agregación.

6.5. Generar conclusión

Si bien este paso se encuentra todavía en estado de desarrollo, podemos dar algunos detalles acerca del mismo:
Las distintas técnicas de agregación utilizadas durante los pasos anteriores del proceso de agregación arrojan distintos resultados con diferentes niveles de fiabilidad. Se hace necesario entonces analizar si los resultados obtenidos por cada una de las técnicas son coherentes entre si, cuales son los mas fiables y cuales son las discrepancias.
Por otra parte, el informe final debe contener dos apartados: "Piezas de Conocimientos obtenidas" y "Posibles Líneas de Investigación". Dentro del primer apartado se describirán los conocimientos obtenidos para los cuales existe evidencia firme que los sustente. Mientras que en el segundo grupo se describirán las conjeturas hechas durante la interpretación de los datos y las preguntas de investigación que no han podido ser resueltas.

7. Estrategia de validación

La fiabilidad y versatilidad del presente método serán validadas mediante comparaciones con otros métodos de agregación. Esto se hará en dos etapas, una primera etapa de laboratorio, donde a un conjunto de datos generados de forma artificial se aplicaran un conjunto de procesos de agregación, para luego comparar los resultados, y una segunda etapa en la cual se buscarán Revisiones Sistemáticas reales en las cuales se haya podido aplicar Meta-Análisis, y se compararan los resultados arrojados por estas contra los resultados estimados mediante el nuevo proceso de agregación. Esta última validación se hará con estudios hechos probablemente en otras ramas de la ciencia, debido a las limitaciones que existen para aplicar un proceso de agregación estándar con los estudios hechos en IS.

8. Conclusiones y Futuras Líneas de Investigación

El trabajo realizado en esta primera etapa de la investigación revela la complejidad de la agregación de estudios empíricos, así como la cantidad de trabajo pendiente. No obstante, a pesar de esto, el objetivo planteado, desarrollar un proceso de Agregación especialmente adaptado para IS, se demuestra factible.
Respecto de los siguientes pasos en la investigación, en este momento se estima prioritario seguir analizando otras ramas de la ciencia para tratar de encontrar mas técnicas de agregación y establecer los niveles de precisión de las mismas. También se procederá en breve plazo a validar el procedimiento propuesto utilizando datos generados de forma artificial, antes de pasar a su validación utilizando datos reales.

Bibliografía

[1] Banker and Keremer; 1989; *Scale economies in new software development.* IEEE Transactions on Software Engineering. (15): 10, pp. 1199-1205.

[2] Shull, F.; Carver, J.; Travassos, G. H.; Maldonado, J. C.; Conradi, R., and Basili, V. R.; 2003; *Replicated Studies: Building a Body of Knowledge about Software Reading Techniques.* Lecture Notes on Empirical Software Engineering. Chapter 2, pp. 39-84. World Scientific.

[3] Hu, Q.; 1997; *Evaluating Alternative Software Production Function*. IEEE Transactions on Software Engineering. (23): 6, pp. 379-387.

[4] Wohlin, C., Petersson, H., & Aurum, A.; 2003; Combining data from reading experiments in software inspections: a feasibility study. (pp. 85-132). World Scientific Publishing Co., Inc.

[5] Juristo, N.; Moreno, A. M., and Vegas, S.; 2004; *Reviewing 25 Years of Testing Technique Experiments*. Journal of Empirical Software Engineering; 9(1 - 2):7-44.

[6] Jorgensen, M.; 2004; A *Review of Studies on Expert Estimation of Software Development Effort*. Journal of Systems and Software. (70): 1-2, pp. 37-60.

[7] Miller, J.; 2000; *Applying Meta-analytical Procedures to Software Engineering Experiments*. Journal of Systems and Software. (54): 1, pp. 29-39.

[8] Kitchenham, B. A.; 2004; *Procedures for performing systematic reviews*. Keele University; TR/SE-0401. Keele University Technical Report.

[9] Goodman C.; 1996*; Literature Searching and Evidence Interpretation for Assessing Health Care Practices;* SBU; Stockholm.

[10] Davis, A.; Dieste o.; Hickey, A.; Juristo, N.; Moreno, A.; 2006; *Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review*; 14th IEEE International Requirements Engineering Conference (RE'06) pp. 179-188

[11] Pickard, L. M.; Kitchenham, B. A., and Jones, P. W.; 1998; *Combining empirical results in software engineering. Information and Software Technology*.; 40(14):811-821.

[12] Hedges, L.; Olkin, I.; 1985; *Statistical methods for meta-analysis*. Academic Press.

[13] Pillemer, D. and Light, R.; 1980; *Synthesizing outcomes: How to use research evidence from many studies*. Harvard Educational Review.

[14] Evidence-Based Medicine Working Group; 1992; Evidence-based medicine. A new approach to teaching the practice of medicine. JAMA, 268(17), 2420-2425.

[15] Noblit, G. W., & Hare, R. D.; 1988; Meta-Ethnography: Synthesising Qualitative Studies. Newbury Park, CA: Sage.

[16] Fairbank L, O'Meara S, Renfrew MJ, Woodridge M, Sowden AJ, Lister-Sharp D.; 2000; *A systematic review to evaluate the effectiveness of interventions to promote the initiation of breastfeeding*. Health Technology Assessment; 4: 1–171

[17] Glaser BG, Strauss AL.;1967; *The discovery of grounded theory: strategies for qualitative research*. New York: Aldine de Gruyter.

[18] Dixon-Woods, M.; Agarwal, S.; Jones, D.; Young, B., and Sutton, A.; 2005; *Synthesising qualitative and quantitative evidence: a review of possible methods*. Journal of Health Services Research and Policy. ; 10(1):45-53B(9).
[19] Yin, R. K. and Heald, K. A.; 1975; *Using the Case Survey Method to Analyze Policy Studies*. Administrative Science Quarterly; 20(3):371-381.
[20] Ragin, C; 1987; *The comparative method: moving beyond qualitative and quantitative strategies*. Berkeley, California: University of California Press.
[21] Straus, S. E. ; Richardson, W. S.; Glasziou, P., and Haynes, R. B.; 2005; *Evidence-Based Medicine*. Churchill Livingstone.
[22] Cochrane; 2007; *Curso Avanzado de Revisiones Sistemáticas*; www.cochrane.es /?q=es/node/198
[23] Burton, A., Shadbolt, N., Rugg, G. y Hedgecock, A.; 1990. *The Efficacy of Knowledge Elicitation Techniques: A Comparison Across Domains and Level of Expertise*. Knowledge Acquisition 2(2): 167-178.
[24] Crandall Klein, B. y Asociados; 1989. *A Comparative Study Of Think-Aloud And Critical Decision Knowledge Elicitation Method*. SIGAR Newsletter, April 1989, Number 108, Knowledge Acquisition Special Issue, páginas 144-146.
[25] Burton, A., Shadbolt, N., Hedgecock, A. y Rugg, G.; 1988; *A Formal Evaluation of Knowledge Elicitation Techniques for Expert Systems: Domain 1*. Proceedings of Expert Systems '87 on Research and Development in Expert Systems IV**.** Pág. 136-145.
[26] Agarwal, R.; Tanniru, M.; 1990; *Knowledge Acquisition Using Structured Interviewing: An Empirical Investigation;* Journal of Management Information System, M.E. Sharpe; Vol. 7 N. 1
[27] Woody, J.; Will, R.; Blanton, J.; 1996; *Enhancing Knowledge Elicitation using the Cognitive Interview;* Expert system with application; Vol. 10 N. 1
[28] Mohagheghi, P., & Conradi , R.; 2004; *Vote-Counting for Combining Quantitative Evidence from Empirical Studies - An Example*. Proceedings of the International Symposium on Empirical Software Engineering (ISESE'04) .
[29] Kumar, R.; 1996; Research Methodology: A Step-by-Step Guide for Beginners. Adisson Wesley.
[30] Creswell, J. 2003. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. Sage Publications.
[31] Marczyk, G., DeMatteo, D., Festinger, D.; 2005; Essentials of Research Design and Methodology (Essentials of Behavioral Science). John Wiley & Sons.
[32] Gurevitch, J. and Hedges, L.V.; 2001; *Meta-analysis: Combining results of independent experiments.* Design and Analysis of Ecological Experiments (eds S.M. Scheiner and J. Gurevitch), pp. 347–369. Oxford University Press, Oxford.
[33] Grimán Padua; 2007; *Propuesta de un proceso de revisión de estudios empíricos en Ingeniería del Software*; IDoESEM.
[34] Lajeunesse, M; Forbes, M.; 2003; *Variable reporting and quantitative reviews: a comparison of three meta-analytical techniques*. Ecology Letters, 6: 448-454.
[35] Thalheimer W. and Cook S.; 2002; *How to calculate effect sizes from published research: A simplified methodology*. A Work-Learning Research Publication.

Program Slicing Based on Sentence Executability

Salvador V. Cavadini
Project EVEREST, Institut National de Recherche en Informatique et Automatique (INRIA)
Sophia-Antipolis, 06902, France
Salvador.Cavadini@sophia.inria.fr

Diego A. Cheda
DSIC, Universidad Politécnica de Valencia
Valencia, 46022, Spain
dcheda@dsic.upv.es

Abstract

We present *point slicing*, a new slicing technique for imperative programs that gives an answer to the question *Which sentences can be executed if sentence* p *is executed?*, very common in program testing, debugging, and understanding tasks and, as far as we know, not directly addressed by other slicing techniques. Point slicing uses a program point as criterion and computes slices by deleting sentences that are proved to be not reachables by executions including the criterion point. We also show how to extend point slicing criterion to a set of program points and how the new technique can be also used to answer to a more precise question: *Which sentences are possibly executed if sentence* p *is executed in a program state satisfying condition* ϕ*?* Because, minimal point slices are, in general, not computable, we provide definitions of safe approximations for each type of point slice.

Keywords: software engineering, program slicing, conditioning, necessary condition for execution.

1 Introduction

Since its introduction, program slicing was found to be useful in diverse software engineering domains. Each of these domains use slices with different properties; for this reason several kinds of slices where defined in addition to the original static slice: dynamic slice, quasi-static slice, (pre/post)conditioned slice, amorphous slice, and abstract slice among others.

A very frequent question in program testing, debugging and understanding activities is *Which sentences can be executed if sentence* p *is executed?* Paradoxically, this question, as far as we know, is not directly addressed by any slicing technique.

In this work we present *point slicing*, a new slicing technique where the slicing criterion is a program point and the slice is composed by program sentences that are possibly executed if the sentence of the criterion is executed; this way, point slicing gives a answer to the above question.

We show how point slicing can be extended to use more than one program point as slicing criterion and also how to answer a more precise question: *Which sentences are possibly executed if sentence* p *is executed in a program state satisfying condition* ϕ*?* All these definitions are given with theirs respective calculi to safely approximate the minimal slices.

This work is organized as follows. The next section provides an overview of the related work. Section 3 introduces much of the definitions that we will use, leaving more specific ones to be introduced when necessary. In section 4 we formally define point slices and provide a calculi to safely approximate them. Section 5 defines two kinds of point slices that use a set of program points as criterion: *weak* and *strong* multipoint slices. Section 6 introduces another extension of point slices: *conditioned* point slices. Next, limitations and future works are discused. Finally, section 8 concludes.

2 Related Work

The term program slice was coined by Mark Weiser in his doctoral thesis [14], he also supplied a precise definition and an algorithm for calculate imperative program slices. Weiser stays that programmers, while debugging, divide code into coherent pieces that usually are not lexically adjacent. These pieces are named program slices. In a general sense, *slicing* is a program transformation which preserves some aspects of the semantics of the original program. Usually, the program transformation is sentence deletion.

In the years following the publication of Weiser's work, researchers found program slices useful for other activities different from debugging –program testing, program understanding, program integration, reuse, program maintenance and reverse engineering, among others [15, 10]–. Some of these activities needs different kinds of programs slices that are similar but not the same to the first definition. This fact originated the development of new slices types: closure non executable slices, forward slices, dynamic slices, quasi-static slices, semantic slices, preconditioned slices, etc. [13, 1, 15]

The novel slice definitions presented in this work are very related to preconditioned slices. When slicing with a precondition, the goal is to detect sentences that can be proved to be dead code when the program is executed with inputs satisfying the precondition.

The idea of slicing with a precondition was introduced in [6] –further developed in

```
S ::= S;S
    | skip
    | id := E
    | if (B) then S else S endif
    | do S' loop

S'::= S;S'
    | S
    | break

E ::= an expression
B ::= a Boolean expression
```

Language Grammar

```
1 if (a=b) then
2    if (a=c) then
3       r:='equilateral'
     else
4       r:='isosceles'
     endif
  else
5    if (a=c) then
6       r:='isosceles'
     else
7       if (b=c) then
8          r:='isosceles'
        else
9          r:='scalene'
        endif
     endif
  endif
```

Program *Triangles*

Table 1: Partial grammar of a simple imperative language and an example program

[3], [7] and [2]–. Precondition-induced dead code is detected by symbolic execution of the program. The initial program state satisfying the precondition is propagated to all points in the program and then at each conditional program branch a test is made to check if it is possible to infer the condition's Boolean value from the corresponding program state. The same approach was adopted in [11] with the addition of an automatic theorem prover to aid in the automation of condition checking. An equivalent method was proposed in [5] where strongest postcondition calculus is used to propagate the initial program state through the program. A less precise method for compute precondition slices is introduced in [12]. The technique avoids the use of theorem provers by applying abstract interpretation to reason about the effects of the precondition. This allows full automated preconditioned slicing at the price of precision.

3 Preliminary Definitions

We will work with a small imperative programming language (Table 1). Let the sentences of the program be numbered from one to the total number of sentences in the program and the unique exit point of the program be labeled with *end*. The state of a program execution is a set of pairs (x, y) where x is a variable in the program's data space and y its value. Variables not mentioned in the state are considered with an undefined value. Then the pair $\langle \sigma, n \rangle$ means that the sentence n of the program will be executed in a state $\sigma = \{(x_1, y_1), (x_2, y_2), \ldots\}$.

Definition 3.1 The **execution trace** of program S with input set σ_1 is the sequence $\tau^S_{\sigma_1} = \langle \sigma_1, 1 \rangle, \ldots, \langle \sigma_m, end \rangle$. Given the predicate ϕ, $\mathbf{T}^S_{[\phi]}$ denotes the **set of executions traces** produced by executing S with input sets satisfying ϕ.[1]

[1]Notice that, for the sake of simplicity, non termination is not taken into account.

Example 3.2 Let S be the program Triangles (Table 1). Then:

- The execution trace for $\{(a,2),(b,2),(c,3)\}$ is:

$$\begin{aligned}\tau^S_{\{(a,2),(b,2),(c,3)\}} \quad = \quad & \langle\{(a,2),(b,2),(c,3)\},1\rangle, \langle\{(a,2),(b,2),(c,3)\},2\rangle,\\ & \langle\{(a,2),(b,2),(c,3)\},4\rangle,\\ & \langle\{(a,2),(b,2),(c,3),(r,'isosceles')\},end\rangle\end{aligned}$$

- The set of execution traces for $[a = 1 \wedge a < b \wedge b = c]$ is:

$$\mathbf{T}^S_{[a=1\wedge a<b\wedge b=c]} \quad = \quad \{\tau^S_{\{(a,1),(b,2),(c,2)\}}, \tau^S_{\{(a,1),(b,3),(c,3)\}}, \tau^S_{\{(a,1),(b,4),(c,4)\}}, \dots\}$$

- If predicate ϕ is $false$, then there is no input set that can satisfy it, therefore $\mathbf{T}^S_{[false]} = \emptyset$

Definition 3.3 We will say that $\tau^S_{\sigma_1}$ **reaches** sentence -program point- p, noted $\tau^S_{\sigma_1} \twoheadrightarrow p$, if and only if $\tau^S_{\sigma_1} = \langle\sigma_1,1\rangle,\dots,\langle\sigma_i,p\rangle,\dots,\langle\sigma_m,end\rangle$. If p is not member of any pair of the sequence $\tau^S_{\sigma_1}$, we will write $\tau^S_{\sigma_1} \not\twoheadrightarrow p$.

Definition 3.4 We will say that $\mathbf{T}^S_{[\phi]}$ **never reaches** p, noted $\mathbf{T}^S_{[\phi]} \not\twoheadrightarrow p$, if and only if $\forall\sigma : \big((\sigma:\phi) \Longrightarrow \tau^S_\sigma \not\twoheadrightarrow p\big)$.

Example 3.5 Let S be the program Triangles (Table 1) then: $\mathbf{T}^S_{[a=b=c]} \twoheadrightarrow 3$, $\mathbf{T}^S_{[a=b=c]} \not\twoheadrightarrow 5$, $\mathbf{T}^S_{[true]} \twoheadrightarrow 1$ and $\mathbf{T}^S_{[false]} \not\twoheadrightarrow 1$.

The *necessary condition for execution* (***nce***) [4] of a program sentence is a condition on the program input set that is true each time the sentence is executed, i.e. if the point -sentence- of interest is reached by the control flow of the program then the initial input set satisfied the ***nce*** of the point. Formally:

Definition 3.6 The **necessary condition for execution** of sentence p in program S, noted $\boldsymbol{nce}(S,p)$, is a precondition that must be true for every input set σ_1 such that $\tau^S_{\sigma_1} \twoheadrightarrow p$. In other terms: $(\tau^S_{\sigma_1} \twoheadrightarrow p) \Longrightarrow \sigma_1 : \boldsymbol{nce}(S,p)$.

Example 3.7 Let S be the program Triangles (Table 1), $(a \neq b)$ is a ***nce*** for the program point 6. And $(a \neq b \wedge a = c)$ is a stronger one.

Naturally, the set of inputs satisfying a ***nce*** of a sentence includes the set of inputs satisfying the *necessary and sufficient condition* for the execution of the sentence and this set includes the set of inputs satisfying the *sufficient condition* (a.k.a. *path condition*) for the execution of the sentence. For example, a necessary condition for execution for `b:=1` in the program of Table 2 can be $(b = 0)$ while its path condition is $(c \leqslant 9 \wedge b = 0 \wedge a < 0)$ and its necessary and sufficient condition is $(c \leqslant 9 \wedge b = 0 \wedge a - (10 - c) < 0)$. Appendix A contains a detailed calculi for ***nce***.

4 Slicing with a Program Point

A program slice, as defined by Weiser in [14], preserves the behavior of the original program w.r.t. a set of variables at a specified program point. In other words, given a program S, a set V of variables and a program point p, the slice of S w.r.t. *(V,p)* is *S'*, a subprogram of S such that the executions of S and *S'* are indistinguishable w.r.t. the values of variables in V at the point p.

```
if (a<10) then
   x:=0
else
   x:=1
endif;
do
   if (c>9) then
      break
   else
      if ((a<0)&&(b=0)) then
         b := 1
      else
         c := c + 1;
         a := a - 1
      endif
   endif
loop
```

Table 2: Program with a *do-loop* structure

For our pourposes, we will say that a slice or *reduction* is a new program obtained by deleting zero or more sentences from the original program:

Definition 4.1 A program $S' = s'_1; \ldots; s'_n$ is a **reduction** or a **slice** of $S = s_1; \ldots; s_n$, noted $S' \preceq S$, if and only if $\forall i = [1, n] : (s'_i = s_i) \vee (s'_i = \texttt{skip})$.

We will define *preconditioned slice* [6, 3] as follows:

Definition 4.2 Given the program $S = s_1; \ldots; s_n$ and the predicate ϕ, then the **preconditioned slice** of S w.r.t. ϕ, noted $\lhd(S, \phi)$, is a program $S' = s'_1; \ldots; s'_n$ such that $\forall i = [1, n] : s'_i = \chi(\mathbf{T}^S_{[\phi]} \not\twoheadrightarrow s_i, s_i)$.

where χ is a function to replace program sentences by `skip` sentences; let ϕ be a predicate and s a sentence then function $\chi(\phi, s)$ returns `skip` if ϕ holds, s otherwise.

The definition stays that a preconditioned slice of program S w.r.t. the condition ϕ is S', a reduction of S where zero or more sentences were changed to `skip` because they can not be part of any execution that starts with an input set satisfying ϕ.

Example 4.3 Let S be the program Triangles (Table 1), then $\lhd(S, a \neq c)$ can be obtained by changing to `skip` those sentences in ***italics***.

In this section we introduce *point slicing*, a technique that uses a program point as criterion and computes a slice with the following property:

Property 4.4 Let S be a program and S' its *point slice* w.r.t. the program point p, then: $(\tau^S_x \twoheadrightarrow p) \Longrightarrow \tau^S_x = \tau^{S'}_x$

This is, S and S' are two programs such that if the set x of inputs produces an execution trace τ^S including sentence p, then executiong S' with the same inputs will result in the same execution trace. A definition for this slice is:

Definition 4.5 Given the program $S = s_1; \ldots; s_n$ and the program point p, then the **point slice** of S w.r.t. p, noted $\lhd^\bullet(S, p)$, is $S' = s'_1; \ldots; s'_n$ where $\forall i = [1, n] : s'_i = \chi\big(\forall x : (\tau_x^S \twoheadrightarrow p \Longrightarrow \tau_x^S \not\twoheadrightarrow s_i), s_i\big)$.

Example 4.6 In the Taxation Calculation Program (Table 3), sentences in *italics* must be changed to `skip` in order to obtain a point slice w.r.t. sentence 38. Underlined sentences must be changed to `skip` to get a point slice w.r.t. sentence 4. Notice that ***nce*** for sentence 38 is $((\neg blind) \wedge (\neg married) \wedge (age < 65))$ and that for 4 is $(75 > age >= 65)$.

A simple approximation to this slice can be the set of sentences that result from the union of the set of sentences that reaches –in the control flow graph– the point of interest and the set of sentences that are reached from this point; however, a better –i.e. smaller– approximation can be obtained if we use semantic information of the slice point. We can reinforce the slicing process with the information about the initial conditions that the program inputs must satisfy to reach the point of interest. Unfortunately, the exact computation of these initial conditions is, in general, undecidable, thus we must approximate them.

4.1 Safely Approximating Point Slices using Preconditioned Slicing and the Necessary Condition for Execution

The ***nce***, in conjunction with preconditioned slicing, can be used to safely approximate minimal point slices. The approximation is possible because it can be proved that if an execution of program S with inputs satisfying $nce(S, p)$ does not reach the program point n then any execution reaching p do not reach n. This allows us to say that $\lhd(S, nce(S, p))$ is a safe approximation of $\lhd^\bullet(S, p)$ or, more formally, that $\lhd^\bullet(S, p) \preceq \lhd(S, nce(S, p))$.

Notice that the path condition of a sentence can not be used to safely approximate point slices. This is because path conditions are too strong and used in preconditioned slicing may led to the deletion of sentences that can be reached by executions that also reach the sentence of interest. As an example of this circumstance, we can see what happens if we compute a preconditioned slice using the path condition of `b:=1` in the program of Table 2, sentence `x:=1` will be deleted –i.e. changed by `skip`– when, actually, there are executions that include both `x:=1` and `b:=1`. Weakest precondition [8] is neither useful for the purpose of answering the above question because, in presence of loops, there is no guarantee that the computed precondition is the weakest one, thus the same problem as with path condition arises.

5 Slicing with Two or More Program Points

It is possible to extend point slicing to cope with more than one program point. In this section we define two point slices that use a set of points as criterion:

- *weak multipoint slice* is the slice that encodes all the possible traces that reach any of the given criterion program points, and
- *strong multipoint slice* is the slice that encodes all the possible traces that reach all the given criterion program points.

```
1  if (age>=75) then
2     pers:=5980
3  else if (age>=65) then
4        pers:=5720
5      else pers:=4335
       endif
   endif
6  if (age>=65 &&
      inc>16800) then
7     t:=pers-((inc-16800)/2);
8     if (t>4335) then
9       pers:=t
10    else pers:=4335
      endif
   endif
11 if (blind) then
12    pers:=pers+1380
   endif
13 if (married &&
      age>=75) then
14    pc10=6692
15 else if (married &&
         age>=65) then
16       pc10:=6625
17    else if (married ||
            widow) then
18          pc10:=3470
19       else pc10:=1500
         endif
      endif
   endif
20 if (married && age>=65 &&
      inc>16800) then
21    t:=pc10-((inc-16800)/2);
22    if (t>3470) then
23      pc10:=t
24    else pc10:=3470
      endif
   endif
25 if (inc<=pers) then tax:=0
27 else inc:=inc-pers;
28    if (inc<=pc10) then
29      tax:=inc/10
30    else tax:=pc10/10;
31      inc:=inc-pc10;
32      if (inc<=28000) then
33        tax:=...
        else
34        tax:=...  ;
35        inc:=inc-28000;
36        tax:=...
        endif
      endif
   endif
37 if (!blind && !married &&
      age<65) then
38    code:='L'
39 else if (!blind && married &&
      age<65) then
40    code:='H'
41 else if (!blind && !married &&
      age>=65 && age<75) then
42    code:='P'
43 else if (!blind && married &&
      age>=65 && age<75) then
44    code:='V'
45 else code:='T'
   endif endif endif endif
```

Table 3: UK Income Taxation Calculation Program (borro wed from [9]).

First, we introduce the definitions of union and intersection of program reductions, both will be used to approximate multipoint slices. Consider the program $S = s_1; \ldots; s_n$ and two of its reductions $S_1 = s_{1,1}; \ldots; s_{n,1}$ and $S_2 = s_{1,2}; \ldots; s_{n,2}$ then:

Definition 5.1 The **union of reductions** is

$$S_1 \cup S_2 = s'_1; \ldots; s'_n \text{ where } \forall i = [1,n] : s'_i = \chi\left((s_{i,1} = \texttt{skip} \wedge s_{i,2} = \texttt{skip}), s_i\right)$$

Definition 5.2 The **intersection of reductions** is

$$S_1 \cap S_2 = s'_1; \ldots; s'_n \text{ where } \forall i = [1,n] : s'_i = \chi\left((s_{i,1} = \texttt{skip} \vee s_{i,2} = \texttt{skip}), s_i\right)$$

5.1 Weak Multipoint Slicing

Definition 5.3 Given a program $S = s_1; \ldots; s_n$ and a set of program points $Q = \{q_1, \ldots, q_m\}$, the **weak multipoint slice** of S, noted $\lhd^{\bullet}_{\mathcal{W}}(S, Q)$, is $s'_1; \ldots; s'_n$ where $\forall i = [1,n] : s'_i = \chi\left(\forall q \in Q,\ \forall x : (\tau^S_x \twoheadrightarrow q \Longrightarrow \tau^S_x \not\twoheadrightarrow s_i), s_i\right)$.

This kind of multipoint slicing computes a reduction such that if an execution trace induced by an input set x in the original program includes one or more sentences of the slicing criterion, then executing the reduction with x will produce the same execution trace. Symbolically: $(\bigvee_{q \in Q} \tau^S_x \twoheadrightarrow q \in Q) \Longrightarrow \tau^S_x = \tau^{\lhd^{\bullet}_{\mathcal{W}}(S,Q)}_x$

We can define two equivalent safe approximations of $\lhd^{\bullet}_{\mathcal{W}}(S, Q)$ as:

$$\bigcup_{q \in Q} \lhd(S, nce(S, q)) \quad = \quad \lhd(S, \bigvee_{q \in Q} nce(S, q))$$

Using properties of ***nce*** it is possible to prove that making the union of the individual point slices of each point in the criterion set results in the same weak multipoint slice obtained from the point slice using the logical disjunction of a ***nce*** of each point.

Example 5.4 A weak multipoint slice of the program from Table 3 with criterion $Q = \{5, 38\}$ can be obtained by changing to `skip` those sentences that are in ***italics and underlined***.

5.2 Strong Multipoint Slicing

Definition 5.5 Given a program $S = s_1; \ldots; s_n$ and a set of program points $Q = \{q_1, \ldots, q_m\}$, then the **strong multipoint slice** of S, noted $\lhd^{\bullet}_{\mathcal{S}}(S, Q)$, is the program $S' = s'_1; \ldots; s'_n$ where $\forall i = [1,n] : s'_i = \chi\left(\exists q \in Q / \forall x : (\tau^S_x \twoheadrightarrow q \Longrightarrow \tau^S_x \not\twoheadrightarrow s_i), s_i\right)$.

This second kind of multipoint slicing computes a reduction such that if an execution trace of the original program induced by an input set x includes all the sentences of the slicing criterion, then the execution of the reduction with x will produce the same execution trace. Formally: $(\bigwedge_{q \in Q} \tau^S_x \twoheadrightarrow q) \Longrightarrow \tau^S_x = \tau^{\lhd^{\bullet}_{\mathcal{S}}(S,Q)}_x$

We can safely approximate the strong multipoint slice of S w.r.t. Q as:

$$\lhd(S, \bigwedge_{q \in Q} nce(S, q)) \preceq \bigcap_{q \in Q} \lhd(S, nce(S, q))$$

Properties of ***nce*** let us prove that the point slice obtained using the logical conjunction of a ***nce*** of each point in the criterion set is smaller than the, also correct, point slice that can be obtained by intersecting the individual point slices of each point.

Example 5.6 A strong multipoint slice of the program from Table 3 with criterion $Q = \{4, 38\}$ can be obtained by changing to `skip` those sentences that are in *`italics`* `and/or` underlined. This slice correspond to the intersection $\lhd(S, nce(S,4)) \cap \lhd(S, nce(S,38))$. If we use $\lhd\big(S, \bigwedge_{q\in Q} nce(S,q)\big)$ to approximate the point slice, we will get an empty slice because the conjunction between the ***nce*** of 4 and that of 38 is: $\big(((\neg blind) \wedge (\neg married) \wedge (age < 65)) \wedge (75 > age >= 65)\big) \;=\; false$

6 Conditioned Point Slicing

In some situations –e.g. while debugging– we are interested in know which are the statements that can be part of execution traces including a given sentence p that is executed in a program state satisfying certain conditions; in other words, we are interested in a slice that can answer to the question *Which sentences are possibly executed if sentence* p *is executed in a program state satisfying condition ϕ?* Point slicing can be strainforward extended to answer this kind of questions.

Notation: We will note $\tau_x^S \twoheadrightarrow end : \phi$ when the execution of S with inputs x ends in a state satisfying predicate ϕ.

Suppose that $S = s_1; \ldots; s_{i-1}; s_i; \ldots; s_n$ is a program, ϕ a predicate and we are only interested in executions of s_i in a state holding ϕ. We can modify the program to

$$S' = s_1; \ldots; s_{i-1}; \texttt{if } (\phi) \texttt{ then } s_i \texttt{ else skip endif}; \ldots; s_n$$

In this program, s_i is reached only if the execution of s_{i-1} leads to an state satisfying ϕ, thus:

$$\tau_x^{S'} \twoheadrightarrow \mathtt{s_i} \iff \tau_x^{s_1;\ldots;s_{i-1}} \twoheadrightarrow end : \phi \qquad \text{and} \qquad \tau_x^{S'} \not\twoheadrightarrow \mathtt{s_i} \iff \tau_x^{s_1;\ldots;s_{i-1}} \twoheadrightarrow end : \neg\phi$$

Then if we compute the point slice of s_i in the modified program we will get the set of sentences that can be executed if s_i is executed in a state satisfying ϕ. It is possible to avoid the program modification by adding the condition as a new term in the point slice criterion and changing appropriately ***nce*** computation (see Appendix A): $nce(\mathtt{S_1; p : S_2}, p, \phi)$ must be defined as $WP_m\ (S_1, \phi)$.

7 Limitations and Future Work

Like any other technique that uses statically propagated semantic information, the approximation of point slices through preconditioned slicing loses precision when deals with programs with loop structures. In this particular case, loops decrease precision of both: ***nce*** computation, and preconditioned slicing. A common approach to alleviate this problem is to add user-provided information about loops invariants. This information could be used in ***nce*** computation to get stronger preconditions of programs with loops. The problem with this approach is that is not clear what information must be given by the invariant to be useful in the computation of a correct ***nce***.

A safe approach to get more precise point slices approximations in presence of loops is to use the ***nce***, w.r.t. the point of interest, of the sentence located just after the loop: let be the program $S = S_1; L; S_2$ where $S_1 = s_{1,1}; \ldots; s_{n,1}$, $L = \texttt{do } S_3 \texttt{ loop}$, and $S_2 = s_{1,2}; \ldots; s_{p,2}; \ldots; s_{m,2}$ then:

$$\lhd^{\bullet}(S,p) \;\preceq\; \lhd(S_1; L, nce(S,p)); \lhd(S_2, nce(S_2,p)) \;\preceq\; \lhd(S, nce(S,p))$$

Finally, the definitions introduced in this article are for a simple while language; work is required in order to extend the definition of ***nce*** computation to cope with more complex imperative language features such as function calls and pointers.

8 Conclusions

This paper has introduced and formalized *point slicing*, a new technique capable of reduce an imperative program to those sentences that are potentially executed when a given sentence is executed, answering the question: *Which sentences can be executed if sentence* p *is executed?* It has showed that the technique can be extendend to compute slices using a set of points as criterion. In this context, two multipoint slicing definitions are given: *weak* and *strong multipoint slicing*. The paper has also defined *conditioned point slicing*, another extension for the technique. Conditioned point slicing answers a more precise question: *Which sentences are possibly executed if sentence* p *is executed in a program state satisfying condition* ϕ*?*

The provided safe approximations to point slices and its variants are based in preconditioned slices that can be computed by means of different analysis techniques; this gives great freedom in the choice of the precision/computational cost relation of the method to be used to obtain the point slice approximations.

References

[1] David Binkley and Mark Harman. A survey of empirical results on program slicing. *Advances in Computers*, 62:105–178, 2004.

[2] Gerardo Canfora, Aniello Cimitile, and Andrea De Lucia. Conditioned program slicing. In *Harman, M., Gallagher, K. (Eds.), Information and Software Technology Special Issue on Program Slicing*, volume 40, pages 595–607, Amsterdam, 1999. Elsevier Science.

[3] Gerardo Canfora, Aniello Cimitile, Andrea De Lucia, and Giuseppe A. Di Lucca. Software salvaging based on conditions. In *ICSM '94: Proceedings of the International Conference on Software Maintenance*, pages 424–433, Washington, DC, USA, 1994. IEEE Computer Society.

[4] Salvador V. Cavadini and Diego A. Cheda. The necessary condition for execution and its use in program slicing. In *PROLE 2007: VII Jornadas sobre Programación y Lenguajes*, Zaragoza, España, 2007.

[5] I. S. Chung, W. K. Lee, G. S. Yoon, and Y. R. Kwon. Program slicing based on specification. In *SAC '01: Proceedings of the 2001 ACM symposium on Applied computing*, pages 605–609, New York, NY, USA, 2001. ACM Press.

[6] Alberto Coen-Porisini, Flavio De Paoli, Carlo Ghezzi, and Dino Mandrioli. Software specialization via symbolic execution. *IEEE Trans. Softw. Eng.*, 17(9):884–899, 1991.

[7] Andrea de Lucia, Anna Rita Fasolino, and Malcolm Munro. Understanding function behaviors through program slicing. In *WPC '96: Proceedings of the 4th International Workshop on Program Comprehension (WPC '96)*, page 9, Washington, DC, USA, 1996. IEEE Computer Society.

[8] Edsger Wybe Dijkstra. *A Discipline of Programming*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1997.

[9] C. Fox, M. Harman, R. Hierons, and S. Danicic. Backward conditioning: A new program specialisation technique and its application to program comprehension. In *IWPC '01: Proceedings of the 9th International Workshop on Program Comprehension*, pages 89–97, Washington, DC, USA, 2001. IEEE Computer Society.

[10] M. Harman, S. Danicic, Y. Sivagurunathan, and D. Simpson. The next 700 slicing criteria. *2 nd UK workshop on program comprehension (Durham University, UK, July 1996), M. Munro, Ed.*, 1996.

[11] Mark Harman, Rob Hierons, Chris Fox, Sebastian Danicic, and John Howroyd. Pre/post conditioned slicing. In *ICSM '01: Proceedings of the IEEE International Conference on Software Maintenance (ICSM'01)*, page 138, Washington, DC, USA, 2001. IEEE Computer Society.

[12] B. Kim Jang-Wu Jo I. Chung, B.-M. Chang. Abstract program slicings. In *Twentieth IASTED International Conference on Applied Informatics*. ACTA Press, 2002.

[13] F. Tip. A survey of program slicing techniques. *Journal of programming languages*, 3:121–189, 1995.

[14] Mark Weiser. *Program slices: formal, psychological, and practical investigations of an automatic program abstraction method.* PhD thesis, University of Michigan, Ann Arbor, 1979.

[15] Baowen Xu, Ju Qian, Xiaofang Zhang, Zhongqiang Wu, and Lin Chen. A brief survey of program slicing. *SIGSOFT Softw. Eng. Notes*, 30(2):1–36, 2005.

Appendix

A Computing Necessary Condition for Execution

Although *true* is a valid ***nce*** for any program point, more precise –i.e. stronger– ***nce*** are desireable for preconditioned slicing. In this section we provide a definition for the computation of ***nce***. In this definition, we will use function $WP_m\ (S, \phi)$ that must provide a precondition of S that has the following fundamental property:

$$\tau^S_{\sigma_1} \twoheadrightarrow end : \phi \Longrightarrow \sigma_1 : WP_m\ (S, \phi)$$

where $\tau^S_{\sigma_1} \twoheadrightarrow end : \phi$ says that the execution of S with inputs σ_1 ends in a state satisfying predicate ϕ.

The definition of ***nce*** for each basic language construct is the following:[2]

- **Sequence**:

$$\begin{aligned} nce(S_1;\mathtt{p};S_2, p) &= WP_m\ (S_1, true) \\ nce(S_1;Sp, p) &= WP_m\ (S_1, nce(Sp, p)) \end{aligned}$$

[2]**Notation:** we write Sp to mean that the point p is included in the secuence S of sentences.

- **Conditional**:

$$\begin{aligned} nce(\texttt{if (b) then } Sp \texttt{ else } S, p) &= b \wedge nce(S_p, p) \\ nce(\texttt{if (b) then } S \texttt{ else } Sp, p) &= \neg b \wedge nce(S_p, p) \end{aligned}$$

- **Loop**:

$$nce(\texttt{do } Sp \texttt{ loop}, p) = \gamma(mod(S_p), nce(S_p, p))$$

where $mod(S_p)$ is the set of variables that are assigned in S_p and γ is a function that takes a set m of variables, and a property ϕ and returns a property ϕ_I such that: $(\phi \Longrightarrow \phi_I) \wedge (\mathcal{V}(\phi_I) \cap m = \emptyset)$, where $\mathcal{V}(\phi)$ is the set of the variables referenced in ϕ. Thus, considering a general property ϕ composed with elementary expressions E, and logical operators $\wedge, \Longrightarrow, \neg, \vee$ we have that if ϕ does not refer to any variable in w then $\gamma(w, \phi) = \phi$. Symbolically: $(\mathcal{V}(\phi) \cap w = \emptyset) \Longrightarrow \gamma(w, \phi) = \phi$

In the other cases:

$$\begin{aligned} \gamma(w, E) &= true \\ \gamma(w, \phi_1 \wedge \phi_2) &= \gamma(w, \phi_1) \wedge \gamma(w, \phi_2) \\ \gamma(w, \phi_1 \vee \phi_2) &= \gamma(w, \phi_1) \vee \gamma(w, \phi_2) \\ \gamma(w, \neg\phi) &= true \\ \gamma(w, \phi_1 \Longrightarrow \phi_2) &= \gamma(w, \neg\phi_1 \vee \phi_2) \end{aligned}$$

WP_m Computation.

Previously we have defined a way to compute an ***nce*** of a program point using WP_m thus its formalization is given:

- **Skip**: $WP_m\ (\texttt{skip}, \phi) = \phi$
- **Assignment**: $WP_m\ (\texttt{y} := \texttt{exp}, \phi) = \phi_{[y/exp]}$. Where $\phi_{[y/exp]}$ means that each occurrence of y in ϕ was replaced by exp.

- **Conditional**: if $S = \texttt{if (b) then } S_1 \texttt{ else } S_2 \texttt{ endif}$, then:

$$WP_m\ (S, \phi) = (c \Longrightarrow WP_m\ (S_1, \phi)) \wedge (\neg c \Longrightarrow WP_m\ (S_2, \phi))$$

- **Break**: $WP_m\ (\texttt{break}, \phi) = false$
- **Loop**: let $L = \texttt{do } S \texttt{ loop}$ and $B(S) = \{b | b = \texttt{break} \wedge S = s_1; \ldots; b; \ldots; s_n\}$

if $B(S) = \emptyset$ then: $WP_m\ (L, \phi) = false$
if $B(S) \neq \emptyset$ then: $WP_m\ (L, \phi) = \bigvee_{s_j \in B(S)} \gamma(mod(S), WP_m\ (s_1; \ldots; s_{j-1}, \phi))$

Reconocimiento biométrico en aplicaciones de E-Government. Análisis de confiabilidad / tiempo de respuesta.

José I. Carri [1], Ariel Pasini [2], Patricia Pesado [3], Armando De Giusti [4]
{jicarri,apasini,ppesado,degiusti}@lidi.info.unlp.edu.ar

Instituto de Investigación en Informática LIDI (III-LIDI)
Facultad de Informática – UNLP

Abstract

An analysis of the use of biometric recognition in E-Government applications is presented.
In particular, this paper discusses the use of digital fingerprints, analyzing the recognition time and the system reliability in function of the increasing number of users.
The systematic study of the recognition response in function of the DB size is presented together with an analysis of the recognition reliability, considering the information loss levels in the digitalization -"scanning"- of the user's fingerprints.
Finally, previous studies are composed with the requirements of an Internet-based WEB service to analyze a projection of the attainable response times with different numbers of users.

Keywords: *E-Government, Biometric Recognition, Fingerprints, Reliability.*

Resumen

Se presenta un análisis de la utilización de reconocimiento biométrico en aplicaciones de E-Government.
En particular se discute el empleo de huellas digitales, analizando el tiempo de reconocimiento y la confiabilidad de los sistemas utilizados en función de un número creciente de usuarios.
Se presenta un estudio sistemático del tiempo de respuesta del reconocimiento en función del tamaño de la BD y asimismo un análisis de la confiabilidad del reconocimiento, considerando niveles de pérdida de información en la digitalización ("scanning") de la huella del usuario.
Por último se componen los estudios anteriores con los requerimientos de un servicio WEB basado en Internet, para analizar una proyección de los tiempos de respuesta alcanzables con diferente número de usuarios.

Palabras Clave: *Gobierno electrónico, reconocimiento biométrico, huellas digitales, confiabilidad.*

VI Workshop de Ingeniería de Software y Bases de Datos

[1] Becario III-LIDI. Facultad de Informática UNLP.
[2] Jefe de Trabajos Prácticos. Facultad de Informática. UNLP.
[3] Profesor Titular. Facultad de Informática UNLP/ Profesional CIC.
[4] Investigador Principal CONICET. Profesor Titular D.E. Facultad de Informática UNLP.

1. Introducción

En la actualidad el problema de la identificación personal se ha convertido en un desafío para los sistemas de seguridad, por el desarrollo mismo de la tecnología y sus aplicaciones "a distancia" tales como E-Government, E-Commerce o E-Learning [1], [2], [3].

Tecnologías como las tarjetas de ingreso, las tarjetas bancarias, las claves de acceso vía InterNet o cualquier combinación de usuario-password presentan puntos débiles en la seguridad, que pueden resultar críticos según la clase de aplicación en la que estén involucradas [4]. De hecho la identificación fehaciente de la persona en tiempo real es un objetivo complejo al que se enfoca el reconocimiento biométrico [5].

Las técnicas biométricas utilizan características fisiológicas o de comportamiento de las personas para identificarlas: las técnicas de reconocimiento de rostro, de huellas dactilares, de iris, de retina y de la geometría de la mano son las más reconocidas para analizar las características fisiológicas [6], mientras que el reconocimiento de firma y el reconocimiento de voz son las más utilizadas para analizar características de comportamiento [7].

Un sistema biométrico puede ser utilizado para verificación o identificación de una persona. En el caso de verificación se lo utiliza para certificar su identidad: la persona debe primero identificarse con algún otro método y luego se verifica la identidad con una técnica biométrica [16].
En el caso de la identificación, directamente el sistema indica cual es la identidad de la persona. Cada técnica tiene sus características particulares, pero todas tienen dos etapas: registro y verificación o identificación.
El registro en el sistema es el entrenamiento para identificar a una persona. En principio la persona provee su identificación, con algún tipo de documentación, y luego se expone a un dispositivo de adquisición de características (dependiendo de la tecnología utilizada), luego esa información se codifica, se asocia a la identidad de la persona y se almacena. El proceso de registro es muy importante ya que de ese momento en adelante las características físicas de la persona quedarán asociadas a la identidad de la misma.
La etapa de verificación consiste en identificar a una persona que ya se encuentra registrada y validar su identificación adquiriendo las características biométricas y comparándolas con las almacenadas en el sistema.
En este caso la comparación de las características se realiza de una contra una. En la identificación, el sistema directamente adquiere las características biométricas y compara contra toda la base en busca de un patrón que coincida con el adquirido.

A continuación se da una breve definición de las diferentes técnicas biométricas:

Reconocimiento de rostro:
Identifica a las personas a través de los rasgos faciales, tratando de copiar el comportamiento de los humanos al identificar la imagen de un rostro dentro de un conjunto de imágenes preestablecido [8]. Nos permite poder identificar a una persona sin interferir en sus actividades. Es decir no sólo nos permite verificar la identidad de una persona con el consentimiento de la misma, sino que podemos identificarla sin contacto físico.
La técnica tiene un espectro amplio de aplicaciones tanto para verificación como para identificación, pero principalmente son utilizadas en sistemas de vigilancia y seguridad.

Reconocimiento de iris:

Se basa en el color de los distintos anillos que rodean la pupila del ojo.
El iris es una fuente muy importante de características biométricas, ya que tiene aproximadamente 266 características distintas, las cuales se forman en el octavo mes de gestación y permanecen inalterables a lo largo de toda la vida de la persona.
Este reconocimiento se puede aplicar tanto para verificación como para identificación.
Tanto en el proceso de registro como el de identificación se debe contar con la colaboración de la persona ya que se trata de un proceso invasivo que consiste en escanear el ojo con un dispositivo de alta calidad y luego armar un sistema de coordenadas con las características del iris.
Se trata de una tecnología de alto costo, con la que se desarrollan sistemas utilizados para acceder a lugares de alta seguridad [9].

Reconocimiento de retina:
Consiste en capturar y analizar los patrones de las venas de la parte trasera del globo ocular.
El patrón ocular es único de cada persona, y son normalmente estables a lo largo de la vida aunque enfermedades como diabetes, glaucoma o presión alta pueden modificarlo.
El proceso de registro e identificación se realiza con dispositivos muy complejos que requieren que la persona adopte posiciones especiales, lo cual hace que sea la técnica biométrica más invasiva [10].

Reconocimiento de la geometría de la mano:
La mano tiene aproximadamente 96 características por las cuales se puede diferenciar a una persona como por ejemplo el ancho de la palma de la mano, la longitud de los dedos, la distancia entre los dedos, la distancia entre nudillos, etc.
La técnica es útil para la verificación de identidad pero no es lo suficientemente confiable para la identificación [11].
Si bien las manos se mantienen invariantes a lo largo de la vida de una persona, pueden sufrir variaciones debido a cambios ambientales o naturales.

Reconocimiento por huellas digitales:
Se extraen características desde distintos ángulos y sectores del dedo y se almacenan. Las huellas dactilares son inalterables a lo largo de la vida de una persona pero lastimaduras, humedad, cicatrices o suciedad pueden alterarlas.
Se trata de una de las tecnologías más empleadas y en las que se han desarrollado diferentes dispositivos de bajo costo que permiten generalizar su aplicación [12] [13] [14].

Reconocimiento por firmas:
Se utiliza para verificar la identidad de las personas midiendo las características de escritura de su firma.
Al realizar una firma la persona ejecuta una serie de movimientos que contienen una información biométrica única, como el ritmo personal, la aceleración y la presión.
Se trata de una técnica muy difundida que en algunas ocasiones se combina con el encriptado de datos, para obtener mayor seguridad en operaciones sobre InterNet. De hecho muchos países han adoptado legislaciones relacionadas con firma digital, enfocando la utilización de esta técnica básica.

Reconocimiento por voz:
Una persona realiza una combinación de factores fisiológicos al hablar que resultan únicos de cada persona.
Este tipo de reconocimiento utiliza estas características para determinar la identificación de una persona, para lo cual se requiere un proceso de "aprendizaje" para determinar su patrón de voz, en el cual se digitaliza la expresión de los fonemas básicos y se genera un patrón o "template" para futuras comparaciones [15].

2. Análisis de la utilización de la técnica de huellas dactilares

La elección de la técnica a utilizar no es una tarea trivial, ya que algunas son netamente invasivas y pueden ser rechazadas por los usuarios/empleados.
Cada método biométrico mencionado anteriormente tiene sus ventajas y desventajas, según la aplicación que se haga de él y las personas que utilicen los equipos. Ningún método es infalible. Aparte de la precisión y la seguridad de los equipos biométricos, hay que considerar también otros factores, como la facilidad de uso, la aceptación por parte del usuario, el mantenimiento y el costo.
El reconocimiento mediante huellas digitales tiene algunas ventajas sobre otras técnicas biométricas, fundamentalmente porque la tecnología de implementación es más simple, notoriamente más barata y las huellas además son imborrables, cuantificables y únicas.

Las huellas digitales son características exclusivas de los primates.
En la especie humana se forman a partir de la sexta semana de vida intrauterina y no varían en sus características a lo largo de toda la vida del individuo. Son las formas que adopta la piel que cubre las yemas de los dedos.
Están constituidas por líneas que forman salientes y depresiones. Las salientes se denominan crestas papilares y las depresiones surcos interpapilares. En las crestas se encuentran las glándulas sudoríparas. El sudor que éstas producen contiene aceite, que se retiene en los surcos de la huella, de tal manera que cuando el dedo hace contacto con una superficie, queda un residuo de ésta, lo cual produce un facsímil o negativo de la huella.
Cada persona en el mundo tiene sus propias huellas digitales, diferentes a las de cualquier otra persona. Son diferentes en cada dedo en ambas manos e incluso entre gemelos idénticos.
Pero aunque cada uno tiene huellas digitales únicas, hay patrones básicos que siempre se encuentran y ayudan a clasificar las huellas digitales: en un dedo aparecen 7 puntos característicos que se repiten indistintamente para formar entre 60 y 120 puntos. A estos puntos se les llaman minucias, término utilizado en la medicina forense que significa "punto característico".
Además de las minucias, las huellas dactilares contienen dos tipos especiales de rasgos llamados puntos *core* y *delta*. Estos puntos son referidos como los puntos de singularidad de una huella dactilar.
El punto *core* es definido como el punto más alto en la línea curva más interior. Este punto es generalmente usado como punto de referencia para la codificación de minucias. El delta es el punto donde las líneas se dividen o se apartan.

Todas las características mencionadas permiten una digitalización y representación simbólica reducida de la imagen de la huella digital, con muy buena confiabilidad para su identificación sin necesidad de hacer una comparación "píxel a píxel" de las representaciones digitales [17].

3. Verificación de identidad en sistemas de E-Government

Es el interés de los gobiernos avanzar en la modernización de sus sistemas de información llevando sistemas centralizados a sistemas que pueden ser utilizados a través de la Web, ya que esta migración agilizará la utilización de los mismos y permitirá un mayor control y auditabilidad.

En general la utilización de tecnología para el acceso del ciudadano a los servicios propios de la gestión (Municipal, Provincial o Nacional) así como los mecanismos que permiten la participación directa del ciudadano en la toma de decisiones (o en la formación de opinión para la toma de decisiones) se denomina "Gobierno electrónico" o E-Government.

Estos mecanismos de inclusión digital del ciudadano requieren la verificación de las personas que tienen acceso a información de carácter sensible o que están habilitadas para la realización de operaciones concretas (expedientes, solicitudes, votaciones).

En general el grado de seguridad requerido alientan el empleo creciente de técnicas biométricas de reconocimiento. En particular al migrar aplicaciones a sistemas Web se hace crítico el empleo de técnicas (y tecnologías) confiables para el acceso a las funcionalidades que se soliciten.

Si bien no es excluyente, el empleo de técnicas de reconocimiento biométrico como las huellas digitales prometen dar respuesta en numerosos casos de E-government. Este proceso está acompañado de una disminución de los costos de los equipos y un incremento en la accesibilidad a los sistemas de comunicación de alta velocidad en las ciudades [18].

4. Caso de Estudio

Se presenta el análisis de cuatro casos en los que existe la necesidad de validación de la identificación de la persona mediante el reconocimiento de huellas dactilares: acceso a archivos confidenciales, emisión de sufragios, modificación de documentación crítica y obtención de certificados de supervivencia en el cobro de pensiones y jubilaciones

Acceso a archivos confidenciales: la protección de archivos, restringiendo el conocimiento del contenido de los mismos a sólo un grupo de usuarios autorizados, plantea un problema de seguridad que puede ser resuelto a través de la identificación del usuario por sus huellas dactilares. Estos usuarios sólo deben posar el dedo sobre el scanner de huellas dactilares para restaurar el contenido del archivo protegido. Al proteger un archivo se debe indicar qué usuarios pueden tener acceso, los cuales fueron dados de alta previamente en la base de datos junto con sus huellas digitales. El sistema crea un nuevo archivo encriptado conteniendo las huellas de los usuarios seleccionados y el contenido del archivo origen. Para desproteger un archivo, se debe posar el dedo sobre el scanner. Si el archivo contiene la huella del usuario, se desencripta el contenido del archivo origen.

Emisión de sufragios: actualmente los electores se presentan en las mesas de votación con su DNI y el presidente de mesa constata la foto del documento con la de la persona que lo presenta, en muchos casos el DNI está en mal estado o la foto no es clara, quedando la decisión final a criterio del presidente de mesa. Una opción para lograr una verificación de identidad precisa, es la utilización de una técnica biométrica. El reconocimiento por huellas dactilares parece ser lo más adecuado ya que se trata de una técnica sencilla, eficiente, de reconocimiento rápido y poco

invasiva. Para utilizar esta técnica en Urnas Electrónicas que incluyan una Terminal de Autoridades desde donde se verifica la identidad, se puede adicionar el sensor de huellas dactilares a la Terminal para que luego que el presidente de mesa ingrese el DNI del elector, le solicite al mismo que coloque su dedo pulgar derecho sobre el sensor, de esta manera el sistema podrá comparar las muestras obtenidas del sensor con la información asociada al DNI de la persona, pudiendo verificar su identidad. Es posible evitar el ingreso del DNI y que el sistema identifique directamente a la persona a través de sus huellas dactilares, pero esta tarea demoraría más tiempo ya que tendría que comparar la muestra contra todas las muestras de las personas del padrón. El proceso de verificación de electores puede también utilizarse independientemente de la urna electrónica, realizándose el proceso electoral de forma manual.
La flexibilidad de las comunicaciones ha llevado a que las personas puedan realizar reuniones a distancia, pero cuando llega el momento de tomar decisiones críticas, que requieren de una votación, debe garantizarse que quien se encuentre del otro lado de la comunicación sea quien dice ser. Actualmente este tipo de reuniones puede realizarse con diferentes tipos de herramienta sobre Internet donde basta con que la persona que tiene que votar coloque su dígito pulgar sobre el sensor conectado a la computadora o utilice un mouse con lector de huellas digitales para validar su identidad. La información obtenida por el sensor es cifrada bajo métodos de encriptación de clave pública/privada y transmitida hacia el centro de la comunicación donde es recibida, desencriptada y comparada con la información de la persona para verificar su identidad. Este procedimiento puede aplicarse para las votaciones a distancia.

Modificación de documentación crítica: es muy común que las personas trabajen desde diferentes municipalidades, organismos públicos, dependencias, etc., utilizando información centralizada en la Web o en servidores compartidos de acceso remoto. La información allí disponible puede ser de carácter crítica y la modificación de los documentos es un problema, ya que los usuarios de esa información podrían llegar a tomar decisiones erróneas. La tarea de modificar y publicar la información debe realizarse por personas autorizadas. En los casos que la información sea de carácter crítica, es posible la utilización de dispositivos de captura de huellas dactilares para verificar la identidad de la persona que está realizando la modificación.

Obtención de certificados de supervivencia: en la actualidad los jubilados y pensionados que cuentan con apoderados o residen en el exterior, para poder percibir sus haberes periódicamente tienen que presentar un certificado de supervivencia, el cual consiste en que se dirijan a un departamento de policía con su documento y un oficial emita dicho certificado que luego el apoderado presenta ante el ente solicitante. Este trámite podría realizarse a través de un sistema WEB que permita la identificación del interesado por huellas dactilares.

5. El problema de la confiabilidad y el tiempo de respuesta en sistemas Web con identificación biométrica basada en huellas digitales.

La aplicación del reconocimiento de huellas digitales a casos de gobierno electrónico como los mencionados en el punto anterior tiene algunas dificultades concretas, entre las cuales se puede mencionar:

- Normalmente la Base de Datos de usuarios (y por ende de huellas digitales) crece y el problema de su acceso en tiempos razonables escala en forma no necesariamente lineal.

Esto requiere una optimización de los algoritmos de digitalización, representación y reconocimiento de los patrones de la huella, de modo de poder manejar el crecimiento del número de usuarios o bien hacer portable la aplicación, por ejemplo de una ciudad de 10.000 habitantes a otra de 500.000 habitantes.
De hecho entre las soluciones está también la paralelización de los algoritmos de identificación y/o la distribución física de los datos a reconocer, de modo de minimizar el espacio de búsqueda.
En este trabajo se ha realizado experimentación con tamaños de BD desde 300 a 100.000 huellas.

- Las comunicaciones en las aplicaciones de E-Government son un factor crítico. Dado que normalmente lo que podemos esperar de usuarios en una ciudad (o en un área geográfica que abarque varias ciudades) es que tengan conexiones a InterNet de diferente capacidad y considerando que la respuesta de la red WAN que da acceso a los servicios depende de un tráfico que normalmente es muy variable (fundamentalmente cambia con el día y la hora de utilización), modelizar las aplicaciones para asegurar un tiempo de respuesta es muy difícil.
  Más aún, clases de aplicaciones que requieren respuestas en tiempo real desde puntos geográficamente dispersos (por ejemplo voto electrónico con identificación por la huella digital) pueden resultar críticas si no se estiman adecuadamente las condiciones de "caso peor" o si existen fallas en los mecanismos de digitalización previa o en el momento de la operación.

- Por último hay que mencionar que el grado de confianza requerido depende de la clase de aplicación. Una consulta popular no vinculante puede admitir márgenes de error en el reconocimiento seguro de los usuarios mayor que una certificación de una transacción financiera o el registro de una escritura pública. Por esto es necesario estudiar la tecnología a emplear, en el contexto real y analizar detalladamente las posibles causas de falla o error.
  En nuestro caso se estudiaron condiciones de pérdida o alteración de las huellas, para medir el efecto en la identificación efectiva del usuario.

6. Trabajo experimental realizado

Se utilizó un scanner modelo "fs80 de Futronic" de huellas dactilares con sensores de tecnología cmos y un sistema óptico preciso para entregar una imagen de alta calidad de la huella digital. La huella dactilar es iluminada por 4 leds infrarrojos cuya intensidad se ajusta automáticamente a las características de la huella, es decir la intensidad varía si la huella esta húmeda o borrosa por ejemplo.

Para el reconocimiento se utilizó una serie de librerías denominadas VeriFinger. Se trata de un motor de identificación de huellas dactilares diseñado para aplicaciones biométricas concretas. Permite ser utilizado en verificación 1:1 o reconocimiento 1:n, el algoritmo de reconocimiento de huellas utiliza un esquema de identificación a partir de un conjunto de puntos específicos de la huella denominado minucia. Además es tolerante a traslación y rotación de las imágenes de huellas. Utiliza un algoritmo original que permite comparar 30000 huellas por segundo e identificar huellas aún si están rotadas, o trasladadas con sólo 5 a 7 minucias similares (usualmente dos huellas del mismo dedo contienen 20 a 40 minucias similares). No requiere la presencia del centro o delta de la huella en la imagen, y puede reconocer una huella a partir de cualquier parte de la misma. De todas maneras si estas características están presentes, la utiliza para un reconocimiento más confiable. La base de datos está preordenada utilizando ciertas características globales. La comparación es

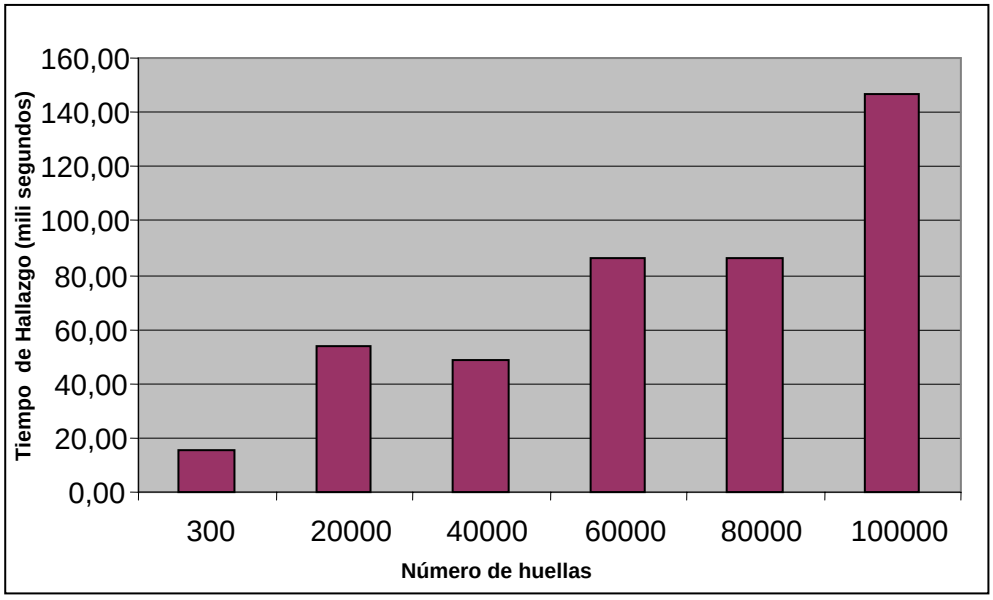
realizada primero contra las huellas almacenadas que contienen similares características globales a la que se está evaluando. Si la comparación contra este grupo no arroja resultados positivos, el próximo registro con características globales similares es seleccionado, y así continúa hasta que el reconocimiento es positivo o hasta que se llega al final de la base de datos. En la mayoría de los casos hay una alta probabilidad de que el reconocimiento exitoso se alcance al comienzo de la búsqueda. Como resultado, la cantidad de comparaciones requeridas para alcanzar un reconocimiento exitoso decrece drásticamente, y consecuentemente, la velocidad de respuesta efectiva es mayor. Enrola por generalización de características a partir de tres imágenes de la misma huella. Cada imagen es procesada y sus características son extraídas. Luego las tres colecciones de características son analizadas y combinadas en una sola colección de características combinadas, que es la que se escribe en la base de datos. De esta manera la minucia enrolada es más confiable y la calidad y confiabilidad del reconocimiento son incrementadas.

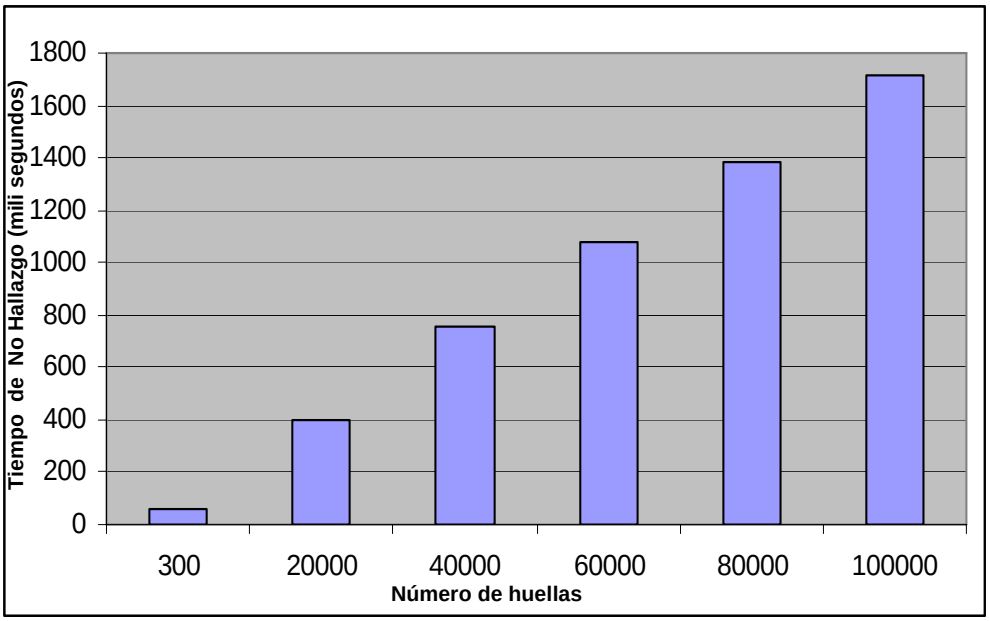
Escalabilidad de los tiempos de respuesta según el volumen de la base de datos

Se almacenaron alrededor de 300 huellas dactilares, luego se replicaron en la base hasta llegar a los 20000, 40000, 60000, 80000, 100000, a las copias se les introdujeron alteraciones para garantizar que en el proceso de identificación las huellas sean únicas.

En la tabla y gráficos presentados se analizan los tiempos de respuesta en el reconocimiento con la BD creciente.

Cantidad de huellas	Tiempo de hallazgo (mili segundos)	Tiempo de no hallazgo (mili segundos)
300	15,16	59
20000	53,79	399
40000	48,91	756
60000	86,42	1076
80000	86,51	1384
100000	146,89	1711





Análisis de alteraciones en la adquisición de la huella dactilar

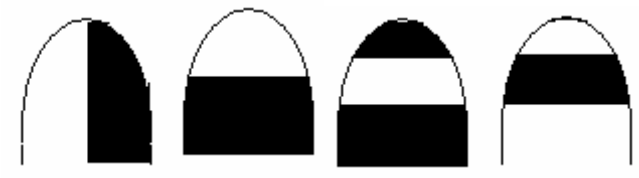
Se simularon una serie de alteraciones que pueden ocurrir en el proceso de adquisición de la huella.

- Alteraciones en la digitalización

Se simuló suciedad en la ventana de adquisición del scanner cubriéndola con cinta adhesiva transparente, hasta dos capas funcionó correctamente, luego la calidad de la huella no fue suficiente para la identificación ya que en la imagen se identificaban claramente los límites de la cinta adhesiva que fueron tomados como posibles minucias de la huella a buscar.
Se intentó realizar una identificación a bajas temperaturas, es decir la huella dactilar por alrededor de los cero grados y el scanner no adquirió la huella.

- Alteraciones de las yemas de los dedos

Se cubrió la yema del dedo con marcador de tinta indeleble como se ve en la figura y la huella no fue identificada, luego se realizó la misma prueba con marcadores al agua y se verificó la identificación en los cuatro casos.



Cubriendo la yema del dedo con cinta adhesiva transparente de la misma forma que indica la figura, las huellas fueron reconocidas en todos los casos.

7. Conclusiones y líneas de trabajo futuro

Se ha analizado la utilización de reconocimiento biométrico (en particular huellas digitales) en aplicaciones de E-Government, poniendo énfasis en los tiempos de respuesta para considerar la escalabilidad de las aplicaciones con el número de usuarios.
Se han estudiado casos de pérdida de información y su efecto sobre el porcentaje de identificación positiva de usuarios, de modo de considerar la confiabilidad de los procesos en función de la posible degradación de la información digitalizada.
Actualmente se trabaja analizando la dependencia del tiempo de respuesta según las características de los links de comunicación y la dinámica del tráfico sobre los mismos (sobre todo enfocado a InterNet).
También se estudia la paralelización y eventual distribución de la Base de Datos de huellas para mejorar los tiempos de respuesta, según la clase de aplicación de E-Government de interés.

8. Bibliografía

[1] Caballero Sybil Lorena. "Prácticas emergentes: la ciberdemocracia, las telecomunidades de conocimiento y los telecentros como alternativas para el desarrollo". CDC, jan. 2005, vol. 22, no. 58, p.97-114. ISSN 1012-2508.
[2] Brunet C., De Lafontaine J. y Schilling K. "Tele-Education in Engineering Using a Virtual International Laboratory". Innovations 2003 – World Innovations in Engineering Education and Research. 2003.
[3] Cabello R. y otros. "EMERGE: A European Educational Network for Dissemination of Online Laboratory Experiments". Innovations 2004 – World Innovations in Engineering Education and Research. 2004.
[4] Reid Paul. "Biometrics for Network Security". Prentice Hall 2004.
[5] Chirillo John y otros. "Implementing Biometric Security". Wiley Publishing 2003.
[6] Woodward J.D. Jr. y otros. "Biometrics". McGraw-Hill Osborne Media.
[7] Liu Simon and Silverman Mark. "A Practical Guide to Biometric Security Technology". IEEE Computer Society, IT Pro – Security.
[8] Biometric Consortium. http://www.biometrics.org/.
[9] International Biometrical Group. http://www.biometricgroup.com/.
[10] Areitio Javier, Areitio Teresa. "Análisis en torno a la tecnología biométrica para los sistemas electrónicos de identificación y autenticación". http://www.redeweb.com/_txt/630/52.pdf
[11] "Information Security - Challenges in Using Biometrics". Statement of Keith A Rhodes Chief Technologist Applied Research and Methods. http://www.gao.gov/new.items/d031137t.pdf
[12] Arsaute Gabriel Adrián, Tutores: Nasisi Óscar Herminio, Martín Marcelo. "Reconocimiento de características en huellas dactilares para la identificación humana". Universidad Nacional de San Juan. Facultad de Ingeniería. Instituto de Automática. 1997
[13] Beavan Colin. "Huellas dactilares. Los orígenes de la dactiloscopía". Ed. Alba. 1990.
[14] Arrieta Angélica, Marín José, Sánchez Luis García, Romero Luis, Sánchez Lázaro Ángel, Batista Vivian. "Gestión y Reconocimiento Óptico de los Puntos Característicos de Imágenes de Huellas Dactilares". Universidad de Salamanca.
[15] Grasso M, Finin Tim. "Integración de tareas en ambientes de reconocimiento de voz multimodales". 1999
[16] Jain L. C., Halici U., Hayashi I., Lee S. B.. "Intelligent biometric techniques in fingerprint and face recognition". 1999.
[17] Lee H. C., Gaensslen R. E. "Advances in fingerprint technology". 1994
[18] Srinivasan V. S., Murthy N. N. "Detection of singularity point in fingerprint images. Pattern Recognition". Vol 25, pp. 139-153. 1992.

Simulación del Modelo de Transacciones Electrónicas Comerciales CLUSTER_EB

Carlos María Chezzi
CIDISI (UTN- FRCON)
Salta 277, Concordia, Entre Ríos
carlos_chezzi@frcon.utn.edu.ar

and

Ana Rosa Tymoschuk
Ariel Villamonte
CIDISI (UTN- FRSF)
Lavaise 610 Santa Fe,
anrotym@ceride.gov.ar
villaariel@hotmail.com

Abstract

The design of computer systems for sites of electronic business can be done with varied configurations of hardware and software resources and communications. Basically it has to consider the clients' requirements as regards efficient levels of attention when they make their commercial transactions. Because of that, it is convenient to have computer technological and communication platforms that assure the quality of the service with a dynamic surfing. The evaluation of the efficiency of these systems, preferably before its implementation, guides the decisions about the investments and the potential benefits. For the planning of the capacities of computer systems there are analytical methods such as the queue and queue networks theory, and their basis to obtain metrics of performance are fundamental laws and algorithms. In the case of sites of electronic business the application of these methods has certain limitations because of the complexity of these systems, mainly due to the clients' behaviour. An advisable alternative is to work with models for simulation that incorporate the representation of complexities and the dynamic analysis of the system behaviour. In this work a study case is stated and solved with the implementation of the modelling methodology for simulation of electronic transactions B2C (Business to Consummer).

Keywords: *Modelling. Simulation. Performance. Prediction. Electronic business. Study case.*

Resumen

El diseño de sistemas informáticos para sitios de negocios electrónicos puede realizarse con configuraciones variadas de recursos de hardware, software y comunicaciones. Fundamentalmente debe considerar los requerimientos de los clientes en cuanto a eficientes niveles de atención al hacer sus transacciones comerciales. Por ello, es conveniente contar con plataformas tecnológicas informáticas y de comunicaciones que aseguren la calidad del servicio con una navegación dinámica. La evaluación del rendimiento de estos sistemas, preferentemente previa a su implementación, orienta las decisiones sobre las inversiones y los beneficios potenciales. Para la planificación de capacidades de sistemas informáticos existen métodos analíticos como la teoría de colas y de redes de colas, cuyos fundamentos para la obtención de métricas de performance son leyes fundamentales y algoritmos. En el caso de sitios de negocios electrónicos la aplicación de estos métodos tiene ciertas limitaciones por la complejidad de estos sistemas, fundamentalmente debida al comportamiento de clientes. Una alternativa conveniente es trabajar con modelos para simulación que incorporen la representación de complejidades y el análisis dinámico del comportamiento del sistema. En este trabajo se plantea y resuelve un caso de estudio con la implementación de la metodología de modelado para simulación de transacciones electrónicas B2C (Business to Consummer).

Palabras Claves: *Modelización. Simulación. Performance. Predicción. Negocios electrónicos. Caso de estudio.*

1 INTRODUCCIÓN

Las configuraciones de recursos de hardware y software para el diseño de sistemas informáticos han incrementado en eficiencia pero también en complejidad, como computadores con capacidades de multiprocesamiento simétrico, utilización de memorias caché, grupos de computadoras interconectados que trabajan juntas como un sistema unificado, entre otras [1],[3],[10].

Mediante la utilización de redes de alta velocidad para la interconexión de servidores y sistemas distribuidos se configuran plataformas de alto rendimiento, que aseguran la disponibilidad y funcionalidad de los negocios basados en transacciones electrónicas [3],[7].
En consecuencia, se plantean diferentes aspectos de hardware, software, comunicaciones y aplicaciones responsables de implementar las transacciones electrónicas y de este modo, no resulta simple la evaluación de las capacidades de la planificación del sistema completo.
Entre las herramientas de predicción, la teoría de colas es una técnica analítica que presenta cada recurso como una cola y propone ecuaciones para la obtención de métricas de evaluación de capacidades [6].

Pero las configuraciones de sistemas informáticos en general requieren de redes de recursos de diferentes características, y su tratamiento se hace mediante la teoría de redes de colas [9]. Las desventajas de estas técnicas analíticas residen en que no permiten representar en forma simple la complejidad de los sistemas de negocios electrónicos o realizar un análisis dinámico de los mismos.

La alternativa es implementar técnicas de simulación, cuya principal ventaja se debe a la capacidad de representación de sistemas reales, mediante relaciones lógicas matemáticas y la experimentación en diferentes escenarios la transforma en una herramienta de soporte de decisiones [6].

En este trabajo se propone un caso de estudio donde se implementa la metodología de modelización para simulación de transacciones electrónicas de negocios, del tipo cliente-negocio B2C [2],[7]. Se propone un modelo CLUSTER_EB, basado en un clusters de servidores [3], agrupados en niveles de servicios (servidores web, de aplicación y de base de datos). Se plantean las cargas de trabajo sobre la base de tres clases de transacciones electrónicas, como etapa preliminar para el estudio de cargas variadas. Con el análisis de los resultados de las primeras experimentaciones se analiza el "cuello de botella" del sistema. Se proponen mejoras y se obtienen métricas e indicadores tecnológicos y métricas de negocios.

2 MODELIZACION

La teoría de colas se basa en el modelo general de colas formado por un centro de servicios y una cola de espera de aquellos clientes que no pueden ser atendidos. En la figura 1 se muestra un modelo de cola con centro simple de servicio.

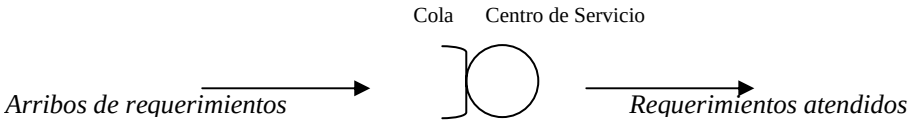


Figura 1. Modelo de Cola

El centro de servicio tiene asignado un tiempo de servicio para atender los requerimientos que arriban según un tiempo entre llegadas. Estos pueden ser atendidos o pasados a cola de espera en caso de que el centro esté ocupado.

En los sistemas informáticos cada uno de los recursos que lo componen es análogo al funcionamiento del centro de servicio con su cola y los requerimientos de usuarios. El tiempo de respuesta de un requerimiento es la suma del tiempo que estuvo en cola y del tiempo de servicio en el centro para su atención. Según la teoría de colas así se determinan los tiempos de residencia del requerimiento en los centros de servicio [6].

El modelo de cola de la Figura 1 tiene dos parámetros: tasa de arribo de los clientes (intensidad de la carga) y requerimiento de servicio promedio de un cliente (demanda de servicio). Con ecuaciones de las leyes fundamentales se pueden obtener medidas de performance tales como utilización del recurso, tiempo promedio del requerimiento en atención (tiempo de residencia), número promedio de requerimientos en cola y velocidad de procesamiento. Es importante destacar que el número promedio de clientes en cola incluye los requerimientos en cola y en servicio [6],[9].

La teoría de colas representa al sistema completo como un solo centro. La teoría de redes de colas permite representar a cada uno de los recursos de los sistemas informáticos, conectados entre sí como una red de cola.

La Figura 2 presenta una red de colas compuesta de una CPU y un Disco. Los requerimientos arriban a la CPU y son atendidos si esta se encuentra desocupada, en otro caso quedan en cola, una vez atendidos por la CPU pasan al Disco. Atendido el requerimiento en el Disco vuelve a la CPU, para finalmente salir como requerimiento atendido.

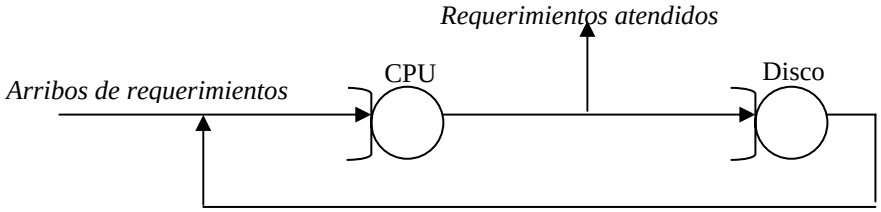


Figura 2. Modelo de Red de Colas

Los parámetros de este modelo son análogos al de teoría de colas con la diferencia de que existen demandas de servicio para cada de recurso de la red. Para la obtención de las medidas de performance de los recursos y del sistema se plantean ecuaciones basadas en Leyes Fundamentales, Ley de Little, Ley de Utilización, Ley del Tiempo de Respuesta y Ley de Flujo Forzado y en algoritmos asociados a cada tipo de modelo [6].

La clasificación de modelos según:

- El sistema que representan: estáticos (no hay cambios del sistema en el tiempo) y dinámicos (se representan los cambios del sistema en el tiempo). Para sistemas dinámicos las variables podrían modificarse continuamente en el tiempo (continuos) o a pasos escalonados (discretos).
- La identificación del Número de Clientes: Abiertos (no se conoce el número máximo de clientes pero sí una tasa de arribos) y Cerrados (se conoce a priori el número total de posibles clientes).
- Carga de Trabajo: Clase Simple (un único tipo de trabajo) y Clases Múltiples (diferentes tipos de trabajo).

La modelización analítica se basa en modelos estáticos, de tiempo continuo, determinísticos o probabilísticos, con requerimientos abiertos y cerrados, para cargas de trabajos simples y múltiples.

Otra de las herramientas de modelización que permite la representación de sistemas informáticos con importantes cantidades de recursos y relaciones de funcionamiento es la simulación, mediante la cual podemos analizar el comportamiento del sistema informático desde un perspectiva dinámica.

3. MODELO CLUSTER-EB PARA SIMULACIÓN

El caso de estudio consiste en la simulación de un proceso de negocio electrónicos B2C, con la aplicación de la metodología basada en el formalismo y la herramienta DEVS JAVA [8] . El sistema informático tiene una configuración basada de capas: la primera capa consiste del cluster de servidores Webs, la segunda, del cluster de servidores de Aplicación y la tercera, del cluster de Base de Datos. La conexión entre capas se efectúa a través de tres redes Lans. Cada LAN se considera una conexión de tipo fullduplex. El arribo de los requerimientos de Internet se hace a través del Router, quien los deriva a los clusters del Sitio de Negocios Electrónicos. Cada grupo de servidores posee un coordinador que toma los requerimientos y los asigna al servidor desocupado.

La base del diseño del modelo de simulación es la teoría de redes de colas donde el sistema informático es considerado un red de colas, cuyos servidores del cluster, redes y router son considerados colas y sus interconexiones, la estructura de la red. Cada recurso tiene asociado una demanda de servicio, que es el requerimiento de servicio promedio por clase de trabajo en el recurso. El modelo es probabilístico, de tipo abierto, y sus entradas son las velocidades de arribos de las múltiples clases de clientes, con la característica de variables aleatorias y distribuciones de probabilidad, que en este estudio se plantea como distribuciones exponenciales.

Las salidas de simulación son las métricas de performance y se basan en las leyes fundamentales y las ecuaciones de la teoría de red de colas [6].

El modelo CLUSTER_EB está representado en un diagrama UML en la Figura 3. Se organiza en dos partes fundamentales: el sistema en estudio y el marco experimental. El sistema se presenta como un acoplamiento de modelos atómicos que representan los recursos informáticos. El marco experimental se forma por los generadores de la carga de trabajo y el transductor que computa las salidas del sistema. Los acoplamientos se describen en la clase ViewableDigraph y cada modelo atómico es parte de la clase ViewableAtomic. A continuación se describen las Clases del modelo:

- *Requerimiento*: representa los requerimientos de los clientes. Contiene un identificador.
- *ClaseRequerimiento*: contiene información sobre la clase a la que pertenece un determinado requerimiento y la ruta que debe seguir el mismo. Por ruta se entiende a la lista de clusters por los cuales debe pasar el requerimiento.
- *GeneradorClase*: genera los requerimientos de una clase determinada siguiendo una distribución de tiempos entre arribos definible.
- *Transductor*: registra los requerimientos generados y completados por clase. Con esta información realiza estadísticas como tiempo promedio de respuesta, throughput, etc.
- *MEClusterRandomEB*: es el marco experimental formado por los generadores y transductores, uno por cada clase de requerimiento, el cluster web, el cluster de aplicaciones y el cluster de base de datos. Esta clase se encarga de configurar los parámetros de los clusters, los generadores, los transductores y las clases de requerimientos.
- *ClusterRandomClase*: representa un cluster. Recibe requerimientos y los procesa. Permite configurar la cantidad de procesadores y sus distribuciones de tiempos de procesamientos para cada clase de requerimiento. Está formado por un administrador de transacciones

(AdministradorTransaccion), un coordinador (CoordinadorRandom) y los procesadores (ProcesadorClase).

- *AdministradorTransaccion*: realiza el seguimiento de cada requerimiento recibido para determinar el destino próximo según la ruta dada por la clase de requerimiento.
- *CoordinadorRandom*: coordina el trabajo de los procesadores y distribuye los requerimientos entre los procesadores, además de recibir los requerimientos procesados desde los servidores y enviarlos a su destino final.
- *ProcesadorClase*: procesa los requerimientos recibidos siguiendo una distribución de tiempo de procesamiento. Permite definir una distribución distinta para cada clase de requerimiento.

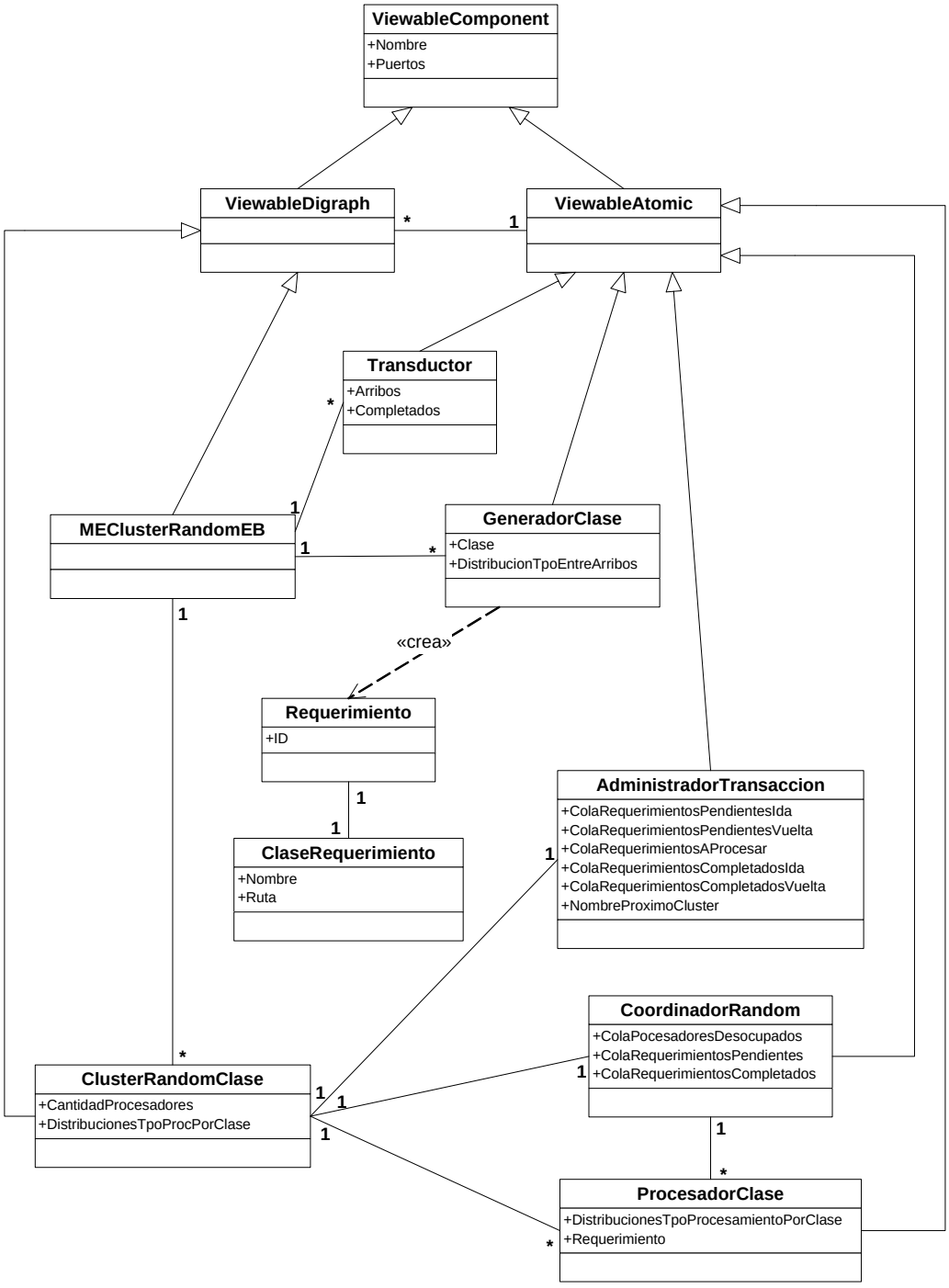


Figura 3. Diagrama UML Modelo CLUSTER_EB

3.2.- Descripción de la Carga de Trabajo.

En la Figura 4 se presenta un modelo de interacción de clientes con tres tipos de sesiones denominadas Clases 1, 2 y 3. La clase 1 visita la página principal (Home), la Clase 2 busca un producto y la Clase 3 hace la adquisición del producto. Estas interacciones dan la funcionalidad del modelo de simulación.

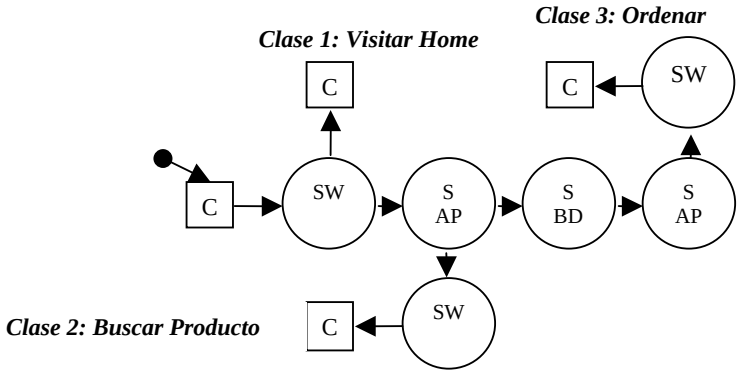


Figura 4: Diagrama de Carga de Trabajo

Donde C: Cliente, SW: Servidor WEB, SAP: Servidor de Aplicación, SBD: Servidor de Base de Datos y ●→: Comienzo.

3.3.- Diseño de experimentos de Simulación.

La simulación se realiza ingresando como entradas los argumentos de la distribuciones exponenciales de cada clase de cliente, obteniendo así los diferentes tiempos medios entre arribos de requerimientos al CLUSTER_EB. Se consideran los tiempos entre arribos para la clase de trabajo 1, manteniendo los tiempos de las otras dos clases constantes. Los niveles de este factor son: 0.06, 0.07, 0.08, 0.09, 0.10 segundos. Cada experiencia de simulación consiste una corrida donde se cambia la semilla de generación de variables aleatorias de 0 a 3 en la herramienta de simulación DEVS JAVA para cada nivel del factor. Cada experiencia demanda un tiempo de simulación de 500 milisegundos [5].

3.4.- Resultados del Modelo CLUSTER_EB.

Para el caso de estudio se propone una plataforma tecnológica con cuatro servidores para el Cluster Web, tres servidores para el Cluster de Aplicación y dos para el Cluster de Base de Datos. Como parámetros del modelo se plantean las demandas de tiempo de servicio por cada clase de trabajo en los servidores del cluster, cuyo detalle se muestra en la tabla 1 y las correspondientes a los dispositivos de comunicaciones se muestran en la tabla 2.

Tabla 1. Parámetros de los Servidores por Cluster para cada clase de trabajo

	Demandas de los componentes del sistema (segundos)			
Clases	Distribución de Tiempos entre Arribos	Cluster Web (4 servidores web)	Cluster Aplicación (3 servidores de aplicación)	Cluster Base de Datos (2 servidores de bases de datos)
1	Constante	0,712		
2	Constante	0,699	0,891	
3	Constante	0,724	0,909	1,333

Tabla 2: Parámetros de los dispositivos de conexión y redes

Dispositivo	Demandas de Servicio (segundos.)
Router	0.0004
LAN 1	0.01546
LAN 2	0.01546
LAN 3	0.01546

El marco experimental de los arribos de requerimientos al sistema está formado por tres generadores cuya descripción se detalla en la tabla 3, siendo variable para la clase 1. Las salidas de simulación son captadas por los transductores.

Tabla 3. Modelo de Intensidad de carga de trabajo

Clase de trabajo	Requerimiento	Distribución de tiempos entre arribos	T: tiempos entre arribos (segundos)
1	Visitar Home	Exponencial	0.06, 0.07, 0.08, 0.09, 0.10
2	Buscar Producto	Exponencial	3
3	Ordenar	Exponencial	15

La hipótesis de simulación es: ¿Cuál es el efecto que tiene la variación de tiempos entre arribos de la clase 1: Visitar el Home en la performance del sitio de negocios?. Para aceptar o rechazar esta hipótesis se simula variando el argumento de la distribución exponencial de la clase 1.

Sobre el escenario planteado se analiza el comportamiento del sistema según la velocidad de procesamiento y el tiempo de respuesta. Los resultados obtenidos muestran tendencias razonables, como se observa en la Tabla 4, con aumento de la velocidad de procesamiento y disminución del tiempo de respuesta según aumenta el tiempo medio entre arribos para los requerimientos de navegación (clase 1: Visitar Home):

Tabla 4. X y R por tiempo medio entre arribos.

Tiempo medio entre arribos: T para la clase 1 (segundos)	Velocidad de Procesamiento X (requerimientos/seg)	Tiempo de Respuesta: R (segundos)
0.06	5.47	90.88
0.07	5.43	76.18
0.08	5.39	71.33
0.09	5.36	64.94
0.10	5.34	60.77

En la Figura 5 se muestra la disminución del tiempo de respuesta por clase a medida que aumenta el tiempo medio entre arribos para los requerimientos de navegación (clase 1: Visitar Home).

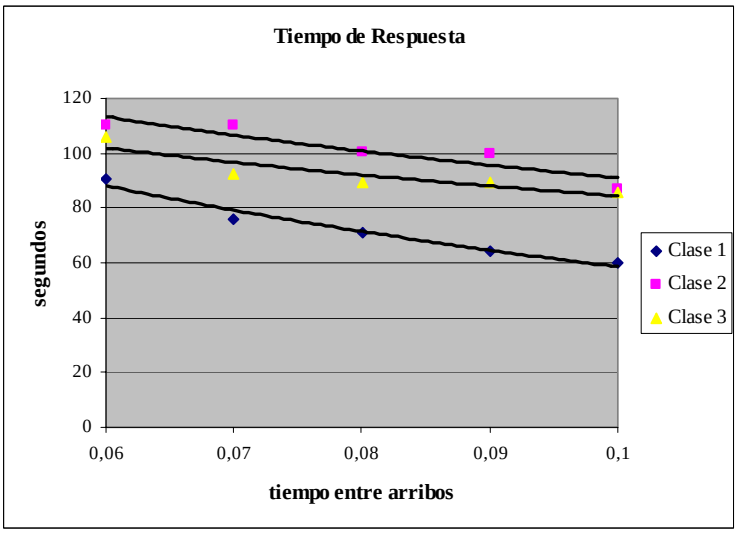


Figura 5: Tiempo de Respuesta por Clase

4 ANÁLISIS DEL CUELLO DE BOTELLA.

Para encontrar el recurso que está limitando el rendimiento del sistema se realiza un análisis de detección del "cuello de botella" y las métricas en esa situación.. Este análisis se basa en las ecuaciones de límites asintóticos de la teoría de redes de colas [6], que incluye la Ley de Utilización según la ecuación:

$$U_K = \lambda * D_k;$$

donde λ es la tasa de arribos al sistema y D_k es la demanda de servicio del recurso k. La utilización tiende al 100 % para una determinada velocidad de arribos λ, llamada de saturación λ_{sat}, cumpliéndose que:

$$\lambda_{sat} = \frac{1}{D_{\max}}$$

La última ecuación indica el límite de productividad como la frecuencia máxima de requerimientos que el sistema puede procesar. En este caso λ_{sat} = 0,75 req./seg.

Según este análisis se calculan las utilizaciones de los recursos del sistema y se muestran en la tabla 5. Se observa que el cuello de botella está dado por los servidores del cluster web, dada principalmente por los requerimientos pertenecientes a la clase 1:

Tabla 5. Utilizaciones por cada recurso del sistema.

Recursos	Utilización (%)
Servidor Web 1	100
Servidor Web 2	99,9
Servidor Web 3	99,95
Servidor Web 4	99,91
Servidor de Aplicación 1	5
Servidor de Aplicación 2	4
Servidor de Aplicación 3	4
Servidor de Base de Datos 1	2
Servidor de Base de Datos 2	2

5.- MÉTRICAS E INDICADORES.

Para evaluar la performance de negocios electrónicos se deben considerar métricas e indicadores a nivel tecnológico y a nivel de negocios.

5.1.- Métricas a Nivel Tecnológico.

En este caso de estudio sólo se platean las métricas tecnológicas de la velocidad de procesamiento y del tiempo de respuesta.

5.1.1. Velocidad de Procesamiento.
X = número de requerimientos de clientes completados por unidad de tiempo.
Unidad: requerimientos/segundo

5.1.2. Tiempo de Respuesta.
R = tiempo medio de respuesta a un requerimiento de cliente.
Unidad: segundos.

5.2.- Indicador del Tiempo de Respuesta (IR).

Si R ≤ 0.23 seg. entonces IR = Sobresaliente
Si 0.23 seg < R < 1.77 seg. entonces IR = Satisfactorio
Si R ≥ 1.77 seg entonces IR = Insatisfactorio

Para la propuesta de los valores del indicador se tomó un ranking de tiempos de respuesta de sitios de negocios reconocidos [4]. El menor tiempo de respuesta es de 0.23 seg y el promedio es de 1.77 seg.

5.3.- Ecuaciones de la Velocidad de Procesamiento y Tiempo de Respuesta

5.3.1. Velocidad de Procesamiento:

$$X_{media} = \sum_{Clase=1}^{3} X_{clase}$$

Ecuación 2: X del sistema

La ecuación 2 muestra que la velocidad de procesamiento del sistema es la suma las velocidades de procesamiento individuales de cada clase.

5.3.2. Tiempo de Respuesta:

$$R_{media} = \sum_{Clase=1}^{3} \frac{R_{clase} X_{clase}}{X_{media}}$$

Ecuación 3: R del sistema

El tiempo de respuesta se calculo de acuerdo a la ecuación 3, donde se observa que no es la suma de los tiempos de respuesta de cada clase o su media aritmética; sino un promedio ponderado en función de las cantidades de trabajos atendidos por clase.

5.4.- Métricas a Nivel de Negocios.

En términos de decisiones es conveniente pensar en un balance entre la performance de la plataforma tecnológica y el monto de la inversión. Por ello se plantea una métrica de negocios en función de la ecuación 4 del punto de equilibrio [2]:

Costo de Inversión = Cantidad de Transacciones Ordenar x Ingreso Neto Promedio por Ventas
Ecuación 4: Punto de Equilibrio

Tomada la métrica tecnológica X para la clase 3 (Ordenar) en un período de tiempo se obtiene la cantidad de transacciones Ordenar, que multiplicada por un ingreso promedio por ventas debe tender al Costo de Inversión para llegar al equilibrio. Un valor menor indica una pérdida y un valor mayor, las ganancias. Es importante tener en cuenta el tiempo en el cual se va a obtener este retorno.

6 PLANTEO DE ESCENARIOS DE SIMULACIÓN.

De acuerdo a los resultados de la tabla 4 y del indicador del tiempo de respuesta la configuración planteada muestra una performance no satisfactoria. Para mejorar la performance se realza un análisis en donde se tienen en cuenta dos escenarios:
Escenario 1: Mínimo Costo de Inversión: se mantienen los parámetros del sistema original y se agrega un servidor al cluster web de las mismas características que el existente.
Escenario 2: Nueva plataforma tecnológica: es una propuesta de mayor inversión, que contempla la actualización de los servidores existentes, duplicando la capacidad de cada uno de ellos y la incorporación de un servidor en cada cluster.
En los dos escenarios se mantiene la configuración de los recursos de comunicaciones.

Tabla 6. X y R por tiempo medio entre arribos para cada escenario.

T (clase 1) (segundos)	Escenario 1		Escenario 2	
	X (requerimientos/seg)	R (segundos)	X (requerimientos/seg)	R (segundos)
0,06	6,80	74,41	13,65	26,46
0,07	6,78	66,94	13,55	10,69
0,08	6,73	60,72	12,78	0,93
0,09	6,73	52,31	11,29	0,55
0,1	6,68	42,97	10,21	0,48

En la Tabla 6 se observa que la alternativa de menor inversión no satisface el indicador del tiempo de respuesta. En el caso del escenario 2 satisface este indicador con cargas de trabajo cuyos tiempos entre arribos T son 0.08, 0.09 y 0.1.

Sobre la base de T = 0.08 se plantea la ecuación 4 del punto de equilibrio:

Costo de Inversión = 0,072 req./seg. x Ingreso Neto Promedio por Ventas $ seg./req.

Con la estimación de los costos de inversión de la plataforma informática del escenario 2 y el ingreso promedio por ventas del negocio se analiza la tendencia al punto de equilibrio.

7 CONCLUSIONES

El trabajo presenta un modelo de simulación aplicado a un caso de estudio simple, con tres clases de trabajo en un sitio web B2C, con una plataforma tecnológica conformada en cluster de servidores web, de aplicación y de base de datos, configurado en tres capas.

El modelo de simulación en DEVS JAVA permite representar la característica aleatoria y de clases múltiples de la carga de trabajo de cualquier sitio de negocios electrónicos. Con la resolución del

sistema con la herramienta de simulación DEVS JAVA se obtienen métricas que se utilizan para generar indicadores del comportamiento del sistema en cuanto al rendimiento desde el punto de vista tecnológico. Estos indicadores se relacionan con métricas de negocios para determinar los beneficios en función de las inversiones y del desempeño del sitio al momento de requerimientos de los clientes.

Los resultados presentados en este trabajo corresponden a un caso simple, previéndose en trabajos futuros la consideración de un número mayor de clases de trabajo y trayectorias aleatorias de los clientes en el sitio de negocios.

Asimismo se tiene en cuenta la necesidad de contar con otras métricas de negocios así como incorporar distintos segmentos de transacciones electrónicas, en vista a la mayor popularidad del uso de Internet para transacciones comerciales.

REFERENCIAS

[1] Alam, Fasihul. McNaught, Ken. Ringrose, Trevor. A comparison of experimental designs in the development of a neuronl network simulation metamodel. *Simulation Modelling Practice and Theory.* 12. Agosto 2004. 559-578.

[2] Chezzi, C.M. Villamonte, A. Tymoschuk, A. R. Modelado y Simulación de Performance de Transacciones Electrónica Comerciales. *Proceeeding IX Workshop de Ciencias de la Computación (WICC 2007).* 443-447. Mayo, 2007.

[3] Faour, A. Mansour, N. Weblins: A Scalable WWW Cluster-Based Server. *Advances in Engineering Software*. Vol. 37. Nº 1. 11-19. Enero, 2006.

[4]http://www.keynote.com/keynote_competitive_research/performance_indices/business_index/business.html

[5] Kleijnen, Jack P. An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research.* 164. Marzo 2004. 287-300.

[6] Lazowska, E. Zahorjan, J. Graham, S. Sevcik, K. "Quantitative System Performance. Computer System Analysis Using Queueing Network Models". Ed. Prentice Hall Inc. (1984). Cap. 1 y 3.

[7] Menascé, D. A., Almeida, V. A. F. *Sacling for E-Business, Technologies, Models, Performance and Capacity Planning*. Ed. Prentice Hall. New Jersey. 2000. Cap. 1, 11, 13.

[8] Nidumolu, S. Menon, N. M. Zeigler, B. Object-oriented Business Process Modeling and Simulation: A Discrete Event System Specification Framework. *Simulation Practice and Theory.* Vol 6. Nº 6. 533-571. Septiembre, 1998.

[9] Puigjamer, R. Serrano, J. J. Rubio, Alicia. *"Evaluación y Explotación de Sistemas Informáticos"*. Ed. Síntesis S.A. Madrid. Cap. 1, 4 y 5.

[10] Wang, P. C. Planning experiments when some specified interactions are investigated. *Computational Statistics & Data Analysis.* 51. Febrero 2007. 4143-4151.

Sistema Colaborativo de Revisión para el soporte de información de contexto en el marco C-INCAMI

María Belén Rivera; Hernán Molina; Luis Olsina

Grupo de I+D en Ingeniería de Software
Departamento de Computación, Facultad de Ingeniería, UNLPam,
Calle 9 esq. 110, (6360) General Pico, La Pampa
Email: belenrs@yahoo.com, [hmolina, olsinal]@ing.unlpam.edu.ar

Resumen

Cualquier organización decidida a implantar programas de medición y evaluación, con el objetivo de mejorar sus procesos y productos, podría beneficiarse si contara con un marco que defina de manera precisa, cada uno de los conceptos involucrados en tales actividades. De esta manera, seria viable la realización de análisis y comparaciones entre los distintos proyectos llevados a cabo y, además, efectuar recomendaciones más consistentes a nivel de organización. Si además de la información necesaria para llevar a cabo dichos programas, se considera el contexto en el cual el proyecto está enmarcado, los resultados finales y recomendaciones obtenidos serán mucho más robustos. La información contextual que se utilice en dichas actividades debería ser acordada. Un sistema de revisión es la clave para que la información de contexto provenga del consenso generalizado entre expertos del área. Se presenta en este trabajo un sistema de revisión que permitirá la discusión y consenso de la información de contexto relevante referida a las actividades que involucra el marco de medición y evaluación C-INCAMI.

Palabras claves: INCAMI, medición, evaluación, contexto, aseguramiento de calidad, revisión colaborativa.

1 INTRODUCCION

En la implementación de proyectos de desarrollo de software, los procesos de medición suponen, junto con los de evaluación, una de las actividades principales previstas en los estándares para el control y el aseguramiento de la calidad[1]. Las organizaciones debieran promover programas de medición y evaluación como parte de la tarea de gestionar la calidad dentro de su línea de producción. Los procesos de medición son fundamentales dado que permiten cuantificar un conjunto de características deseadas acerca de un aspecto específico de algún ente en particular, proveyendo una visión más o menos detallada de su estado o condición. Por su parte, la evaluación interpreta los valores obtenidos en la medición. Para dichos procesos de medición y evaluación es necesario obtener datos cuantitativos, a partir de métricas de atributos de entes y la posterior interpretación de la medida a partir de indicadores. Es numerosa la información existente referida a la definición de métricas e indicadores, sin un claro consenso en cuanto a la terminología. En este sentido consideramos que la Ontología de Métricas e Indicadores presentada en [4] constituye una importante propuesta en el área de gestión calidad y un aporte valioso para las actividades implicadas en dicha gestión.

Las organizaciones decididas a encarar un proyecto de medición y evaluación podrían beneficiarse al contar con un marco que permita definir e instanciar cada una de las actividades y

[1] Algunos estándares ISO tratan sobre calidad, procesos y evaluación: ISO 9126-1:2001 relacionada a la calidad del producto software, ISO 15504:2003 : proceso de calidad, ISO 14598:1998 establece el proceso de evaluación, ISO 15939:2002 define el proceso de medición

conceptos que ellas implican. Si bien hay estudios[2] en el tema, en este trabajo nos enfocaremos en el marco de medición y evaluación orientado a propósitos y centrado en la organización, denominado INCAMI [5]. INCAMI se fundamenta en el método WebQEM (*Web Quality Evaluation Method*) [7], el cual se basa en modelos y métricas de calidad y se centra en la evaluación cuantitativa de características y atributos de entidades. De esta manera, INCAMI puede ser utilizado en el diseño de requerimientos no funcionales, en la selección de métricas para cuantificar los atributos de las entidades involucradas y en la interpretación de los valores correspondientes mediante indicadores, conforme a la necesidad de información de nivel organizacional y de proyecto enunciada. Como base conceptual tiene la Ontología de Métricas e Indicadores mencionada anteriormente. Adicionalmente, ha sido diseñada una herramienta, *INCAMI Project Manager* [5] (referirse a la Figura 1), que puede ser utilizada como soporte en el proceso de medición y evaluación de entidades, en proyectos de software y web de una organización. Hace uso del Catálogo Organizacional INCAMI [3] [6] que contiene las definiciones e instancias de los elementos de información, comunes a toda la organización, necesarios para llevar adelante dicho proceso. Esta herramienta a su vez permite, para un proyecto de medición y evaluación, no sólo guardar los valores finales de las mediciones y evaluaciones sino también los metadatos asociados. Sin este recaudo, el análisis y la comparación de datos podrían no realizarse consistentemente entre los diferentes proyectos de una organización.

El repositorio de medición del catálogo organizacional (ver Figura 1), guarda instancias de métricas y otros conceptos relacionados que han sido concensuados a través de un proceso colaborativo de discusión. Dicho proceso se encuentra soportado por el prototipo del *Sistema de Revisión de Métricas* [1]. Con este sistema de revisión, se garantiza que todos los datos almacenados, relacionados a métricas y al resto de los conceptos, han sido acordados entre expertos y/o interesados de la organización, en el ámbito de calidad y son aquellos útiles a la organización, y se asegura que no habrá definiciones ambiguas o carentes de sentido, en el momento de usarlos en la etapa de medición.

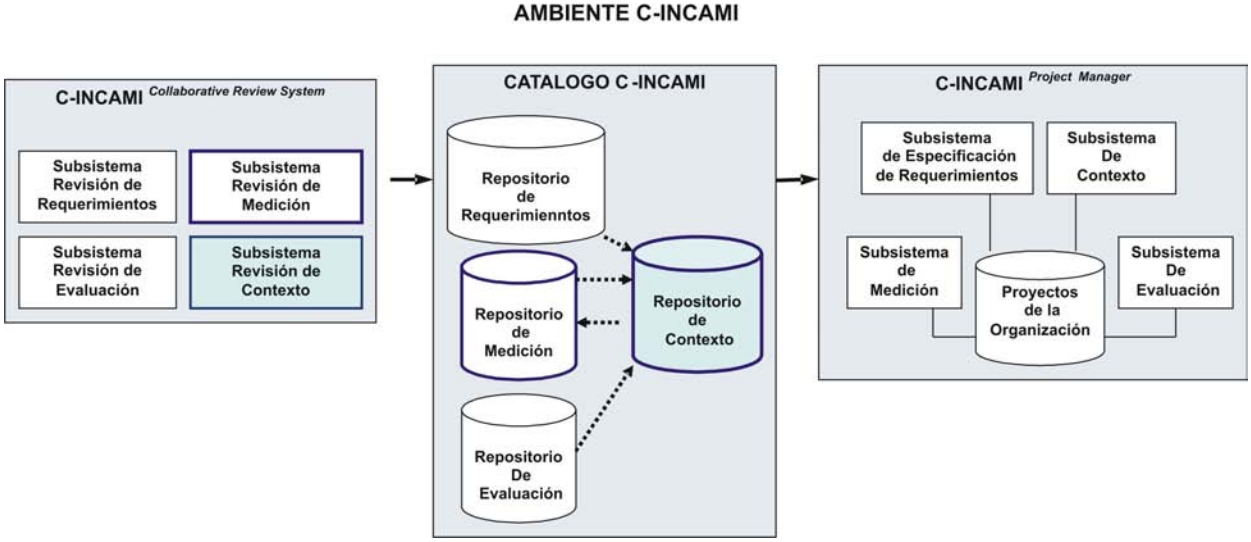


Figura 1: Esquema de los subsistemas principales del sistema que soporta a la Medición y Evaluación basado en contexto

[2] Particularmente GQM (*"Goal Question Metrics")* enfoque para la medición orientada a lograr metas en una organización. Permite a los tomadores de decisión, elegir aquellas métricas que se relacionen a las metas más importantes de los problemas más urgentes. Su desventaja reside en que no parte de una base conceptual bien definida del dominio de métricas e indicadores y el proceso de interpretación de las medidas de las métricas no está claramente definido.

C-INCAMI (*Contextual INCAMI*) constituye una propuesta para anexar, al marco descrito anteriormente, información relativa al contexto de las entidades a medir y evaluar, dado que considera que dicha información es un factor clave en el uso e interpretación de la información a utilizar en los procesos de medición y evaluación. Así como fue necesario que la información referida a métricas, almacenada en su catálogo correspondiente, sea proveniente del consenso general de expertos en el área, será requisito, también en este caso, que la información de contexto en C-INCAMI sea almacenada en sus catálogos correspondientes, luego de pasar por un proceso de revisión similar al del caso del marco INCAMI. Se presenta en este trabajo una extensión del actual sistema de revisión de métricas para incluir los elementos de información relacionados a la información de contexto (los mismos serán analizados en la sección 2). El nuevo sistema, al igual que el de métricas, permitirá discutir, consensuar y finalmente agregar a un catálogo de contexto, la información de contexto relevante según las necesidades de la organización (ver Figura 1).

El resto del artículo se organiza de la siguiente manera: la sección 2 presenta una introducción al marco INCAMI, así como al modelo conceptual de contexto propuesto para C-INCAMI. La sección 3 analiza principios básicos de herramientas colaborativas, describe las piezas a discutir y el proceso de discusión llevado a cabo. Luego, en la sección 4, se ilustrará con un caso práctico el mecanismo de discusión planteado. Los comentarios finales referidos este trabajo serán presentados en la sección 5.

2 INFORMACIÓN DE CONTEXTO EN EL MARCO *C-INCAMI*

2.1 Introducción al marco INCAMI

El marco INCAMI define un conjunto de conceptos relacionados con la medición y evaluación de requerimientos no funcionales en proyectos de software y web, como parte de las actividades de aseguramiento de calidad de una organización. INCAMI está basado en una ontología [4] que define de forma explícita estos conceptos, así como sus atributos y relaciones, y, también en el método WebQEM [7]. Los conceptos principales del marco son Necesidad de Información, Modelo de Concepto, Atributo, Métrica e Indicador. INCAMI es soportado por un catálogo organizacional que contiene instancias de los conceptos definidos en el marco ofreciendo un mecanismo de reuso para permitir una mejor consistencia entre los resultados provenientes de diferentes proyectos en la organización.

INCAMI está estructurado en base a las actividades o fases a realizar en el proceso de medición y evaluación según la metodología WebQEM. La primera fase corresponde a la definición y especificación de requerimientos. Este módulo trata con la definición de la necesidad de información (es decir, el foco de la evaluación) y el diseño de los requerimientos no funcionales, que servirán como guías para las actividades posteriores de medición y evaluación (ver Figura 2). Instancias de conceptos tales como necesidad de información, entidad, atributos, entre otros, son recuperadas del catálogo organizacional.

La fase siguiente corresponde al diseño e implementación de la medición. Este módulo trata con la definición de las métricas que serán útiles para cuantificar los atributos, que en la etapa anterior se identificaron como parte de la especificación de requerimientos, y que son de especial interés en el proyecto, dado que constituyen las características que se medirán para el ente a evaluar, considerando la necesidad de información establecida (es decir, el objetivo final de la evaluación).

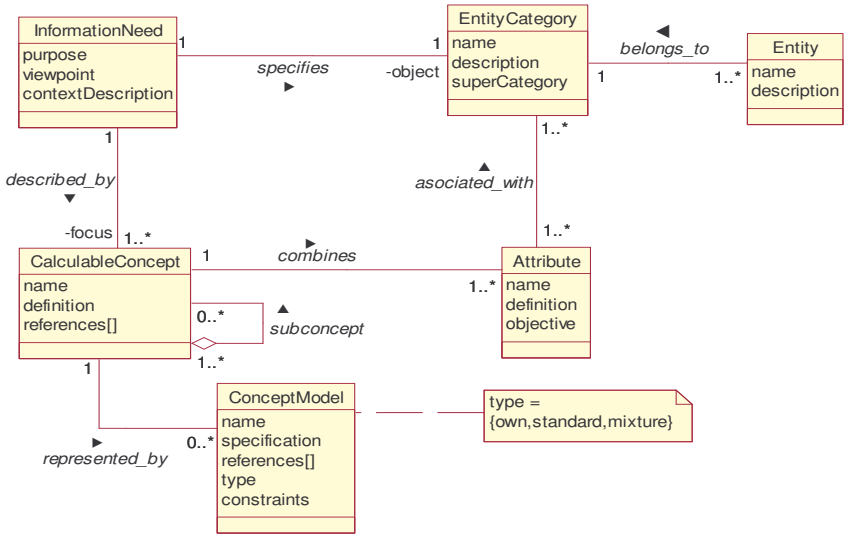


Figura 2: Modelo Conceptual de Requerimientos

Para cada métrica se debe seleccionar la escala y el método de medición o cálculo (según sea directa o indirecta la métrica), tal como se muestra en el modelo de la Figura 3. Estas métricas serán recuperadas del catálogo de métricas de la organización. Finalmente, en la etapa de evaluación, estas métricas deben ser interpretadas a través de indicadores con el objetivo de evaluar o estimar el grado de conformidad que los requerimientos propuestos alcanzaron. Es en este momento cuando deben seleccionarse, del catálogo de evaluación, los indicadores que interpretarán cada métrica que cuantifica a cada atributo correspondiente en el diseño de los requerimientos no funcionales. Los indicadores contienen también una escala y una función o algoritmo a través del cual será posible interpretar el valor de la métrica, con ayuda también de un criterio de decisión, que establecerá umbrales de aceptabilidad al valor obtenido.

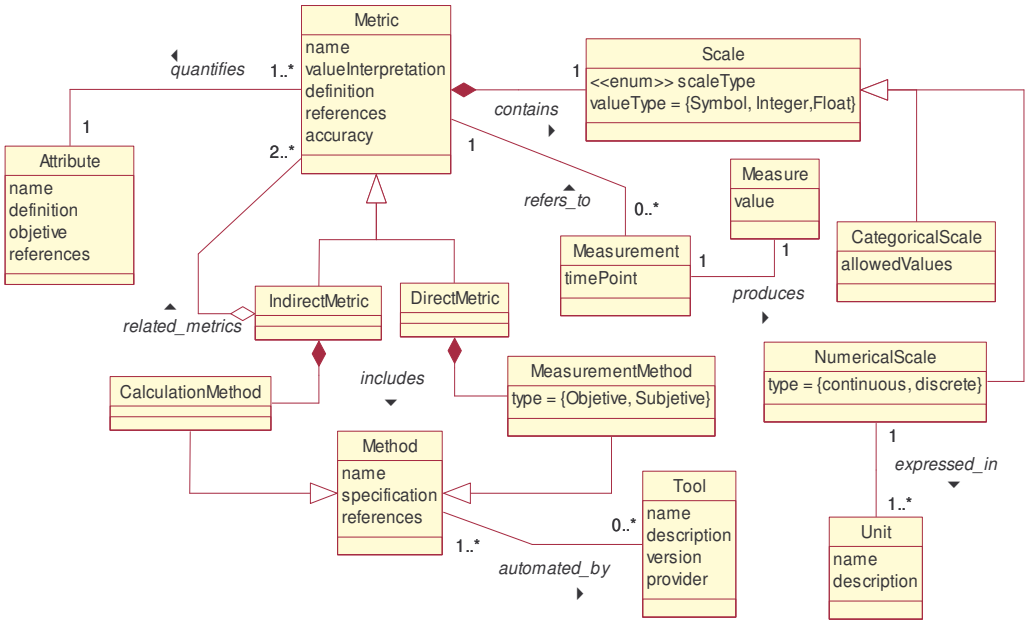


Figura 3: Modelo Conceptual de Medición

2.2 Modelo *C-INCAMI (Contextual INCAMI)*

La propuesta C-INCAMI pretende incorporar al marco INCAMI la información que describa el contexto en el cual se llevan a cabo las actividades de medición y evaluación con la intención de

mejorar la coherencia y consistencia en el uso e interpretación de la información manipulada en tales actividades.

El modelo propuesto está compuesto por dos espacios de información diferentes: el espacio de elementos del dominio (el marco INCAMI existente) y el espacio de elementos de contexto. Los conceptos del dominio de INCAMI son contextualizados estableciendo relaciones con elementos del dominio de contexto. Tal como se expresa en la propuesta C-INCAMI [2], la necesidad de información (*InformationNeed*) incluye un objeto *Context* (ver Figura 4) que contendrá las instancias de las propiedades (*INCAMIContextProperty*) para describir el contexto relevante en el cual se lleva a cabo la medición y evaluación de las entidades especificadas. Por su parte, los elementos contextuales (*ContextualElement*), que identifican cualquier elemento de información del marco INCAMI cuya aplicación es sensible al contexto, estarán asociados a una instancia de un contexto al cual son aplicables, es decir, describen las propiedades de contexto relevantes para la aplicación del elemento contextual en cuestión. Los elementos del marco que llevan información contextual son: modelo de concepto (*ConceptModel*), métrica (*Metric*), indicador global (*GlobalIndicador*), indicador elemental (*ElementaryIndicador*). Es posible entonces, distinguir entre el contexto relevante de la entidad (que, como ya fue expuesto, se especifica en la necesidad de información) y el contexto relevante de un elemento del marco INCAMI (como por ejemplo, una métrica) respecto de su aplicación en un proyecto de medición y evaluación. El primero representa el estado actual del ambiente del proyecto, mientras que la descripción del contexto de los elementos contextuales representa el contexto al cual dicho elemento es aplicable.

Para la modelización del contexto en C-INCAMI se ha utilizado [2], por un lado, un esquema de nombre-valor para representar las propiedades de contexto, mientras que el contexto es modelado como un conjunto o agrupación de estas propiedades. Por otro lado, ha sido diseñada una taxonomía de propiedades de contexto basada en ciertos conceptos claves de la ingeniería de software y web.

Una propiedad de contexto (*INCAMIContextProperty*) (ver Figura 5), tiene como atributos un nombre, una descripción, el peso o importancia relativa de esa propiedad en un contexto dado y un identificador. Una propiedad de éste tipo es considerada como una especialización de un atributo del marco INCAMI. De esta manera, es posible utilizar los conceptos relacionados a medición para cuantificar las propiedades de contexto.

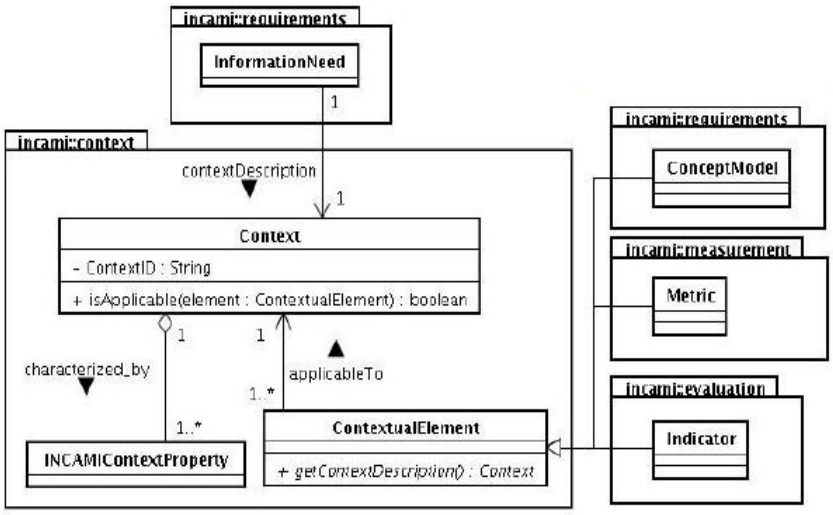


Figura 4: Modelo de C-INCAMI relacionando la información de contexto con el marco INCAMI

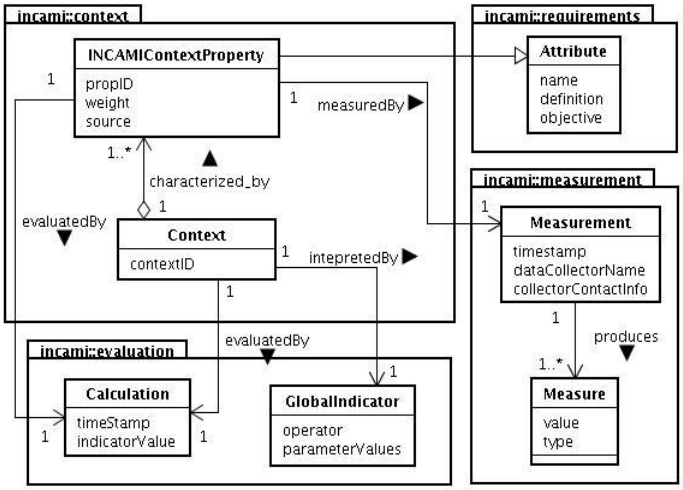


Figura 5**:** Diagrama UML que representa la relación entre el dominio de concepto y el dominio de INCAMI

3 EL PROCESO DE REVISION

A continuación se describirán las piezas de información a ser discutidas en el sistema de revisión de información de contexto así como el proceso mediante el cual se llevará a cabo la discusión. Previamente se mencionan brevemente los principios básicos a tener en cuenta en la construcción de herramientas colaborativas de discusión.

3.1 Principios básicos para herramientas colaborativas de discusión

Es importante, y son motivo de continuos estudios[3], que los procesos colaborativos de discusión estén soportados por herramientas apropiadas de CSCA (*Computer Supported Collaborative System)*, que permitan un fluido desarrollo de las actividades de argumentación en ambientes Web.

Los principios que deben ser considerados al momento de la realización de herramientas, basadas en procesos de discusión, pueden resumirse de la siguiente manera [8]:

- Es importante que los usuarios revisores dispongan de elementos gráficos (como los iconos de acuerdo y desacuerdo) para enmarcar el tenor de sus comentarios.
- El ambiente de discusión debe contemplar la posibilidad de visualizar al mismo tiempo la información a discutir y los comentarios realizados. La navegación entre los ítems en discusión y sus respectivos comentarios no debe alejar al usuario a otra pantalla, mostrando ambos contenidos en forma separada.
- Los revisores deben dedicar a la revisión el máximo de tiempo que les sea posible, sin que ello signifique exigencias extras a su labor.
- Las herramientas que se provean deben ser fáciles de usar e intuitivas, para que usuarios no expertos se sientan familiarizados con las mismas.

Para este tipo de herramientas de discusión, el soporte tecnológico es fundamental. Permite que se automatice todo el proceso y de esta manera agiliza aún más las interacciones del mismo. En el desarrollo de la herramienta aquí presentada han sido considerados como pautas de diseño los principios antes mencionados.

3.2 Definición de las piezas a discutir

Con respecto a la pieza de información a discutir y de acuerdo a las dos situaciones descriptas en la sección 2.2, es posible identificar dos tipos de piezas a discutir.

[3] En especial, hemos considerado el trabajo relacionado a procesos de revisión colaborativa llevado a cabo por el grupo KMI (*Knowledge Media Institute),* pertenecientes a The Open University,U.K.

El primer caso corresponde a una propiedad de contexto utilizada para describir cualquier contexto. Para esta situación, la pieza a debatir será la definición de la propiedad de contexto (básicamente, su nombre, tipo y descripción), además de la métrica que la cuantifica y la escala y método de la misma. Esta propiedad de contexto será discutida, y si alcanza un consenso entre los distintos revisores, se almacenará en su catálogo correspondiente. Luego, será utilizada cuando se tenga que importar, de dicho catálogo (el de contexto) para describir el contexto de algún proyecto de medición y evaluación (como parte de la necesidad de información del mismo), o para describir el contexto de aplicabilidad de algún elemento contextual.

La otra posible pieza de información a debatir es la información de contexto asociada a un elemento contextual (como por ejemplo una métrica), es decir, a qué contexto es aplicable. Esta pieza deberá contener las instancias de propiedades de contexto (que previamente fueron discutidas y almacenadas para el caso anterior) incluyendo además sus posibles valores, el peso o importancia de dicha propiedad en el contexto actual y el elemento contextual al cual se refiere.

Cualquiera sea el caso de la pieza de discusión, cuando alcancen el consenso deberán ser almacenadas en sus correspondientes catálogos. Las propiedades debatidas en el primer caso, serán almacenadas en el catálogo de contexto, mientras que la información de contexto debatida para los elementos contextuales, será almacenada en el catálogo respectivo del elemento (requerimientos, medición o evaluación). De esta manera, tanto las propiedades de contexto como los elementos contextuales estarán disponibles para ser utilizados durante el desarrollo del proceso de medición y evaluación llevado a cabo por la organización.

3.3 Proceso de Discusión

El sistema de revisión que será aplicado a la información de contexto hace uso de mecanismos colaborativos asincrónicos para llevar a cabo el proceso de discusión, por lo tanto, los usuarios pueden participar libremente en los momentos que deseen.

Cabe destacar que el sistema será accedido por diferentes categorías de roles, a saber: el de administrador, el de autor, y el de revisor (ver Figura 6).

- El administrador es el rol del usuario que debe controlar si ha recibido alguna pieza de información a discutir, que, como fue analizado en la sección anterior, pueden ser propiedades de contexto o bien elementos contextuales con características de contexto. Debe verificar la validez de la misma con las instancias de propiedades existentes (para no discutir las ya incorporadas al repositorio). Es responsable de la administración, configuración y seguimiento del ambiente de discusión. Tiene a su cargo también conformar y coordinar el grupo de revisores y establecer el cronograma de trabajo. Debe también decidir, a partir de las votaciones efectuadas por los revisores (y del esquema de puntuación establecido), si la información debatida alcanzó un consenso o no y en caso afirmativo debe consolidar el repositorio con la nueva pieza de información acordada en el proceso.
- El revisor es el rol del usuario encargado de llevar a cabo el debate. Es su tarea opinar sobre la información que se ha puesto a discutir. Además tiene la responsabilidad de participar en la votación, lo que permitirá saber al administrador el grado de consenso alcanzado en el debate.
- El autor es quien somete su propuesta (la pieza de información) al administrador. Si la misma es inicialmente aceptada como pieza a debatir, participa entonces activamente en el proceso de discusión a la par de los revisores. El autor no tiene posibilidad de votar, pero sí tiene privilegios de escritura para introducir algún cambio a la información sometida (ateniéndose a las restricciones de las interacciones del proceso).

Es importante resaltar que no consideramos necesaria la participación de un usuario moderador, como suele suceder en otros sistemas de discusión (como por ejemplo los chats, foros en el que el

encargado de supervisar las discusiones es un moderador). Hemos limitado el alcance del sistema a nivel organizacional, por lo tanto, la figura del administrador y el moderador llevarían a cabo funciones muy similares, y entonces se optó por la presencia de un solo rol de administrador que contemple funciones de administración y moderación.

La Figura 6 muestra el proceso de revisión y los roles de usuarios responsables de cada fase del proceso.

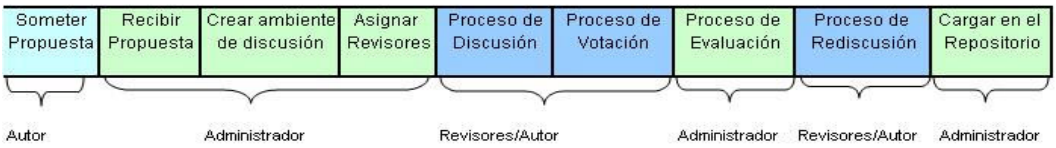


Figura 6: *Etapas del proceso de revisión y roles a cargo.*

Con respecto al proceso de revisión cabe aclarar que es el mismo utilizado para la revisión de información de métricas en [1], contemplando diferencias provenientes del tipo de información que ahora se discute. El proceso comienza cuando una persona (de aquí en adelante *autor)*, la cual se supone que es experta el área de calidad y que con el fin de lograr la revisión de su pieza de información, *somete una propuesta* (ver Figura 6).

Será el administrador quien *reciba la propuesta* (la pieza de información) y en base a sus criterios y conocimientos analizará preliminarmente la instancia de información relacionada a contexto sometida con el fin de detectar si dicha pieza ya ha sido catalogada en el repositorio, o si no merece pasar el proceso de revisión. Si se aceptara provisoriamente, se continúa con la preparación del entorno necesario para la pieza de información. Para esto, el administrador deberá *crear el ambiente de discusión* para que la pieza de discusión pueda ser debatida. Deberá gestionar tareas tales como la fijación de la fecha de inicio y fin del debate, así como también deberá elegir y *asignar a los posibles revisores*, participantes activos en el proceso de discusión. Al ser notificados los revisores de su participación (vía e-mail), deberán responder si aceptan la revisión de la pieza de información.

En el período que abarca el *proceso de discusión* los revisores podrán ingresar al sistema, el cual tendrá la particularidad de dividirse en dos áreas relacionadas (diseñado de esta manera siguiendo las pautas establecidas al inicio de la sección). Por un lado se verá la pieza a discutir, y por el otro, todas las opciones relacionadas con el proceso de discusión (emitir comentario, responder comentario, ver comentarios y votar), a las cuales tendrá acceso el revisor

Cada revisor podrá comentar y responder a comentarios realizados por otros, durante el lapso de tiempo que dure el debate. Podrán utilizar símbolos gestuales para dar más énfasis a la opinión, como los iconos de acuerdo y desacuerdo ().

El resto de los participantes, administrador y autor, siguen desde sus respectivas áreas todo el proceso de discusión, controlando el estado del foro en cada momento. Esta particularidad permite que el administrador, a modo de supervisar el proceso, detecte posibles discrepancias o conflictos no relacionados a la discusión del tema que se está debatiendo.

Dado que el autor es otro de los espectadores del proceso de discusión, podrá observar los comentarios que se fueron sucediendo desde el instante en que comenzó el debate. Además puede defender su pieza de información con el derecho de réplica que se le otorga a partir de la posibilidad que tiene de emitir o responder a un comentario. Podrá analizar el curso que está tomando la información que sometió al principio, es decir si ésta será posiblemente aceptada o rechazada por los revisores. Si el autor prevé que la mayoría de los comentarios generan un rechazo por parte de los revisores, o que no hay un consenso entre ellos, y si cree pertinente que parte de la información pueda ser modificada, entonces solicitará un permiso al administrador. Con este permiso el autor procederá a cambiar los ítems en discusión que crea conveniente y comunicará tal situación a los

revisores para que, en base a las modificaciones efectuadas, establezcan nuevamente sus comentarios y se reanude el debate correspondiente.

Cuando la fecha de finalización ha arribado, la que inicialmente fuera establecida por el administrador, los revisores quedan imposibilitados de continuar debatiendo, pero ahora deberán puntuar la información discutida. Comienza entonces el *proceso de votación*, en el cual cada revisor deberá analizar, según su criterio, la información debatida y asignarle uno de los tres posibles puntajes: aceptado, rechazado o rediscutir con mejoras. El primero significa que el revisor está de acuerdo totalmente en que la pieza debatida sea parte del catálogo organizacional de la empresa, pues ha advertido que es útil, concisa, coherente. La segunda y como es de suponer, implica una total negación por parte del revisor a consolidar dicha pieza. Las razones las puede exponer, para que el autor, y el resto de los revisores conozcan los motivos de tal elección. Por su parte, la tercera y última opción establece que el revisor considera que existen aspectos que pueden mejorarse, y volver a discutirse, para así, pueda lograrse (o no) un posible acuerdo. Durante todo el tiempo que dure la votación, el administrador tendrá acceso a indicadores que le estarán informando sobre el estado de la votación, así también como estadísticas que le indican el grado de conformidad entre los revisores, el grado de desacuerdo y el grado de rediscusión.

Cuando todos los revisores han finalizado con la votación, el administrador deberá *evaluar la discusión*, teniendo en cuenta las puntuaciones establecidas por los distintos revisores. Así, el administrador, no solo podrá observar los valores dados por cada revisor, sino también que el sistema le advertirá los porcentajes resultantes:

- Si más del 80% de los revisores acordaron aceptar la pieza de información, la misma pasa al repositorio.
- Si más del 80% de los revisores acordaron rechazar la pieza de información, esta deberá ser descartada
- Si el 50% aprobó y el 50% rechazó, la decisión final es la de rediscutir.
- Si el 100% opta por rediscutir, se rediscute nuevamente, con los cambios sugeridos.
- Se descarta también para otro caso que no se encuadre en lo detallados, y se evidencia que no se ha llegado a un consenso, por lo tanto, no debe guardarse la pieza debatida.

Si el resultado de la votación estableció una *rediscusión* del debate, el administrador deberá establecer tiempo adicional al foro recién finalizado, indicando la cantidad de días que, a partir de la fecha actual, se extenderá la de finalización del foro. Durante la rediscusión, aquellos ítems que fueron determinados como no acordados por los revisores, estarán visualizados con otro color para enfocar la atención del debate en ellos.

Una vez establecida la rediscusión se comunica respectivamente de la situación a cada uno de los participantes y desde ese momento comienza nuevamente a rediseñarse el debate, repitiéndose el mismo procedimiento que se explicó anteriormente. Hay que tener en cuenta que la posibilidad de rediscusión puede darse una sola vez, pues de lo contrario todo el proceso de discusión se convertiría en un ciclo repetitivo, con final incierto.

Para esta etapa, en la votación solo podrá optarse por aceptar o rechazar (por lo comentado recientemente). Nuevamente, en caso de que la pieza de información sea aceptada se habrá logrado un consenso. Cualquiera sea el caso se da aviso mediante notificación automática al autor y a los revisores de lo sucedido.

Finalmente la responsabilidad pasa nuevamente a manos del administrador, quien en caso de que la información debatida fuese aceptada, deberá realizar la *carga en el repositorio* correspondiente de la misma.

4 UN CASO PRÁCTICO

En esta sección se ilustra el proceso de discusión de una propiedad de contexto, tal como se describió en las secciones previas. Para tal fin, consideremos el siguiente escenario: una empresa dedicada al desarrollo de software puede considerar conveniente evaluar la *productividad en el desarrollo* llevado a cabo por algún *proyecto de software*. Para tal entidad y propósito uno de los posibles atributos a incluir como parte del diseño de los requerimientos no funcionales será *tiempo de desarrollo de software.* Mediante la posterior selección de las métricas, que permitan cuantificar los atributos elegidos y de indicadores que interpreten dichos valores, será posible obtener un resultado que satisfacerá la necesidad de información definida inicialmente para el proyecto de medición y evaluación. Además, permitirá la comparación con el resultado de otros proyectos para poder determinar cuál proyecto ha resultado más productivo. Sin embargo, una vez obtenidos los valores, la comparación puede no resultar del todo robusta dado que, por ejemplo, se pueden estar comparando resultados de proyectos de software cuyos ciclos de vida empleados son muy distintos (por ejemplo un proyecto desarrollado con una metodología tradicional en contraparte con una metodología ágil[4]). La falta de esta información puede influir en la interpretación de los resultados obtenidos, es decir, información referida al contexto del proyecto a evaluar. Por tal motivo, es importante para el caso planteado, considerar el *ciclo de vida* como información de contexto en el proyecto de medición y evaluación, tanto en la selección de métricas e indicadores así como en la interpretación de los resultados finales.

A continuación se ilustrará con un caso práctico, para el escenario descrito anteriormente, el proceso de revisión de la definición de la propiedad de contexto *ciclo de vida del desarrollo del software*. La misma se especifica en la Tabla 1, y es la propiedad de contexto finalmente acordada durante el proceso de revisión. Esta propiedad formará parte del catálogo de contexto, la que podrá ser reusada cuando se tenga que importar de dicho catálogo información de contexto para un proyecto de medición y evaluación específico.

Tabla 1. Ejemplo de definición de propiedad de contexto consolidada en el repositorio de contexto

Nombre: Ciclo de vida de desarrollo de software
Descripción: El ciclo de vida para el desarrollo del proyecto de software
Tipo: SoftwareProjectProperty
Métrica
- *Objetivo*: Determinar el tipo de ciclo de vida del desarrollo del proyecto de software
- *Escala* (*Categórica*)
  - *Tipo de Escala*: Ordinal
  - *Valores Permitidos:*
    - *Nombre*: Ciclo de vida tradicional. *Descripción*: Ciclo de vida en el que se especifica de manera rigurosa estrategias de planificación, recursos y procesos llevados a cabo durante el desarrollo.
    - *Nombre*: Ciclo de vida Ágil. *Descripción*: Ciclo de vida con un desarrollo según los principios y prácticas ágiles.
    - *Nombre*: Ciclo de vida ad-hoc. *Descripción*: Ciclo de vida sin un claro establecimiento de estrategias de planificación, procesos y recursos.
- *Método* (*Medición*)
  - *Nombre:* Determinación del ciclo de vida del desarrollo del proyecto de software
  - *Especificación:* El ciclo de vida utilizado es extraído de la planificación del proyecto u otras evidencias.
  - *Tipo:* Sujetivo.

[4] Tal como es el caso de XP: *Extremming Programming*

Una persona ingresa al sistema de *Revisión de Información de Contexto* y en la opción *someter propuesta*[5] adjuntará la información que quiere poner a discusión. El envío de la propuesta se hará a través de un formulario preestablecido que contiene los campos necesarios para el ingreso de la propuesta según sea de cualquiera de los dos tipos posibles de información a discutir (sección 2). Tan pronto el administrador recibe en su área respectiva la información enviada por el autor, procede a cotejarla con las instancias ya existentes en el repositorio correspondiente. Al comprobar que no hay ninguna instancia que se asemeje a lo que se propone, procede a notificarle al autor la validez de su propuesta. La acción a seguir es comenzar a configurar aspectos relativos del debate: fechas de inicio y fin, y además, elegir a los revisores. Tendrá en su interfaz de usuario opciones siempre disponibles para realizar tales acciones.

Una vez que los revisores confirmen su participación, estarán habilitados a iniciar el debate el día establecido. Comenzarán a opinar, por ejemplo, en base a si es correcto, para la propiedad *Ciclo de vida del desarrollo del software,* considerarla del tipo *SoftwareProjectProperty*. Hemos planteado para el ejemplo propuesto, que, inicialmente, el autor sólo consideró como posibles tipos de ciclo de vida, el tradicional, y el ágil. Es bien sabido que las metodologías tradicionales se centran especialmente en la planificación y control del proceso, estableciendo minuciosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Por su parte, las metodologías ágiles otorgan mayor valor a los recursos humanos, prácticas de desarrollo, colaboración con el cliente e iteraciones muy cortas. Sin embargo, un proyecto puede no tener un desarrollo planificado, que se ajuste a alguno de los ciclos propuestos a considerar. Por esta razón, es importante también incluir el ciclo de vida *ad hoc*, definido como aquel que no presenta una estructuración en sus procesos, ni siquiera una identificación de fases o pasos a seguir (siendo más bien una estrategia code and fix). Por tal motivo y ante la crítica de los revisores hacia el autor, por no haber considerado la inclusión de este tipo de ciclo de vida, el autor procede a incluirla, logrando de esta manera el consenso generalizado por parte de los revisores. El administrador, que sigue el proceso de discusión, observa que el mismo se desarrolla adecuadamente.

Para cerrar el proceso en el día establecido como votación, los revisores deberán puntuar a la propuesta como aceptada, rechazada o bien rediscutirla con mejoras. Aquí, se supone que gran parte de los revisores acordaron y entonces calificaron como *aceptada* a la pieza de información, con un total del 90% de aceptaciones. El administrador dispone de un área de seguimiento del proceso de votación y, en este caso, el sistema le notifica la aceptación de la propuesta de información de contexto debatida. Por lo que el administrador procederá a la carga de la instancia de la propiedad de contexto debatida (tal como se describe en la Tabla 1) en el repositorio de contexto.

5 COMENTARIOS FINALES Y CONCLUSIONES

En este trabajo se ha mencionado la importancia de contar con información de contexto en el marco de medición y evaluación INCAMI, y se ha resaltado la ventaja de lograr que esa información de contexto provenga del consenso alcanzado entre varios revisores expertos en el tema. De este modo, dicha información quedará disponible para todas las actividades de medición y evaluación que una organización implementa como parte del aseguramiento de calidad en sus proyectos de software y web.

El sistema colaborativo de revisión propuesto para el soporte de información de contexto en el marco C-INCAMI contiene los módulos básicos para llevar a cabo el desarrollo de un proceso de discusión, contemplando sus instancias, iteraciones y roles que ejecutan las tareas del proceso.

[5] No se muestra la pantalla de la herramienta actualizada por razones de espacio, pero se puede observar el trabajo realizado en [1]

Desde el punto de vista tecnológico, para gestionar potencialmente grandes volúmenes de datos y metadatos referidos a contexto, se ha utilizado una estructuración semántica especificada en lenguaje RDF (*Resource Description Framework)* y RDF Schema, y se ha empleado el servidor SESAME, que es el que también almacena las instancias de métricas consensuadas en el Sistema de Revisión de Métricas [1]. Es importante destacar que especificar la información consensuada en RDF permitirá la posterior consulta de dichos datos y su recuperación en los procesos que así lo requieran, como así también lograr la vinculación con el resto de los datos que la organización utiliza.

Por último, el contar con un sistema de revisión para procesos de medición y evaluación, no sólo a nivel de métricas de atributos sino también de propiedades de contexto, intenta robustecer al ambiente C-INCAMI con el fin de permitir que una organización disponga de un mejor soporte a la toma de decisiones.

REFERENCIAS

1. Baffini, M., Rivera, M.B., Olsina, L.: Sistema Colaborativo de Revisión de Métricas, 3 Workshop de Ingeniería de Software y Bases de Datos, XII CACIC, San Luis, Argentina, 2006
2. Molina H, Olsina L.: Towards the Support of Contextual Information to a Measurement and Evaluation Framework, QUATIC'07 (Quality of Information and Communications Technology), IEEE Compute Science Press, Lisboa, Portugal, 2007.
3. Molina H., Papa F., Martín M. de los A., Olsina L: Semantic Capabilities for the Metrics and Indicators Cataloging Web System. In: Engineering Advanced Web Applications, Matera M. Comai S. (Eds.), Rinton Press Inc., US, pp. 97-109, ISBN 1-58949-046-0.
4. Olsina L., Martín M.: Ontology for Software Metrics and Indicators, Journal of Web Engineering, Rinton Press, US, Vol 2 Nº 4, pp. 262-281, ISSN 1540-9589.
5. Olsina L., Molina H; Papa F.: Organization-Oriented Measurement and Evaluation Framework for Software and Web Engineering Projects, Lecture Notes in Computer Science of Springer, LNCS 3579, Intl Congress on Web Engineering, (ICWE05), Sydney, Australia, July 2005.
6. Olsina, L.; Martin, Ma A.; Fons, J.; Abrahao, S.; Pastor, O.: Towards the Design of a Metrics Cataloging System by Exploiting Conceptual and Semantic Web Approaches, in LNCS 2722 of Springer. Int'l Conference on Web Engineering 2003 (ICWE'03), Spain.
7. Olsina, L., Rossi, G. Measuring Web Application Quality with WebQEM, IEEE Multimedia, 9(4), 2002, pp. 20-29.
8. Summer T., Buckingham Shum, S.: From Documents to Discourse: Shifting Conceptions of Scholarly Publishing. Proc. CHI 98: Human Factors in Computing Systems, (Los Angeles, CA), 95-102. ACM Press: NY. <d3e.open.ac.uk/general/d3e-chi98/>

Software projects effort estimation using “use case points” method in the context of Project Management Body of Knowledge

R. Uzal (*) (**), G. Montejano (*), D. Riesco (*)
(*) Universidad Nacional de San Luis
Ejército de los Andes 950
5700 San Luis - Argentina
gmonte@unsl.edu.ar
(**) Universidad de Buenos Aires

N. C. Debnath
Winona State University
Winona, MN 55987 - USA
ndebnath@winona.edu

Abstract

As a contribution to consider software projects to be risk investments, it is important the full standardization of either the planning methodology (as a subset of project management methodology) and the effort estimation method. We think there is important and abundant evidence backing the convenience of the join use of Project Management Body of Knowledge and “Use case points” method in software project planning and control. Software development teams should share information, as it is important to compare the “ex post” analysis of programming/budgeting estimations against project execution data with “real world” software projects. This comparative activity would include teams into a permanent improvement process.

Keywords: Software engineering economics, PMBOK, use case points method

1. Introduction

Software development projects are investment projects [1]. Every investment has three key characteristics: Expected Return, Risk and Marketability.

Expected return: It refers to the amount of interest, dividends or capital gains that you expect to receive from your investment. (Actual returns may, of course, be quite different.). There is a direct correlation between expected return and risk. We can say: “The higher the expected return, the greater the risk”.

Risk: It is the possibility of losing some, all or more than your initial investment, or the possibility of receiving less return than you expected. Lower risk investments include, for example, government treasury bills. In the case of software development projects, they are usually high risk investments [2] so, investors will expect high returns.

Marketability (or 'liquidity'): This characteristic refers to the possibility of selling or redeeming your investment quickly at or near the current market price. Term deposits are an example of an illiquid investment, since you are not generally allowed to withdraw your money before the end of the term without paying a significant penalty. Several other investments, such as mutual funds or listed securities, are very marketable as they can be quickly sold or redeemed on short notice and at low cost. Marketability is an important factor to be considered when an investor is selecting his/her investments. Software development projects are generally illiquid investments.

Important investors are needed to face important software development projects. An investor will use a variety of figures to evaluate the financial attractiveness of a software development project. He/she (investor) needs to estimate the project’s capital cost, projected earning, annual revenues, expenses, and tax impact. The three primary figures to be used by investors are:

Net Present Value: Net Present Value (NPV) is the sum of all years’ discounted after-tax cash flows. The NPV method is a valuable indicator because it recognizes the time value of money. Projects whose returns show

positive NPVs are considered to be attractive.

Internal Rate of Return: Internal rate of return (IRR) is defined as the discount rate at which the after-tax NPV is zero. The calculated IRR is examined to determine if it exceeds a minimum acceptable return, often called the hurdle rate. The advantage of IRR is that, unlike NPV, its percentage results allow projects of vastly different sizes to be easily compared.

Payback Period: A payback calculation compares revenues with costs and determines the length of time required to recoup the initial investment. A Simple Payback Period is often calculated regardless the time value of money. This figure of merit is frequently used to analyze retrofit opportunities.

As a preliminary conclusion we can say that a software development project needs a well supported "projected cash flow" in order to begin conversations with potential investors. "No investors, no software project"

In order to obtain the required well supported "projected cash flow", high quality effort estimation method and a consistent planning methodology are needed. In this paper we strongly suggest the join use of Project Management Body of Knowledge (PMBOK) by Project Management Institute (PMI) [3] and "Use Case Point" method [4] [5] [6] [7] in the context of software development planning and control.

2. The "use case points" method

A brief presentation and any reference of "use case points" method as described by Schneider, G. and Winters, J. [4] is developed through this paper. This estimation procedure requires the possibility to count the number of transactions in each use case. A transaction is an event occurring between an actor and the system, the event being performed entirely or not at all.

There are four steps included in "use case points" method:

1. The actors in the use case model are grouped as *simple*, *average* or *complex*. A simple actor represents another system with a defined API; an average actor is another system interacting through a protocol such as TCP/IP; and a complex actor may be, for example, a user (human being) interacting through a graphical user interface or a web-page.
A weighting factor is assigned to each actor group:

• Simple: Weighting factor 1
• Average: Weighting factor 2
• Complex: Weighting factor 3

The total *unadjusted actor weight* (*UAW*) is calculated counting the number of actors in each group, multiplying each subtotal by its specified weighting factor, and then adding subtotals.

2. Each use cases is also categorized as *simple*, *average* or *complex*, depending on the amount of transaction included in a specific use case. We must also consider the transactions in alternative flows. Included or extending use cases are not considered. In general terms, a simple use case has 3 or fewer transactions; an average use case has 4 to 7 transactions; and a complex use case has more than 7 transactions. A weighting factor is assigned to each use case category:

• Simple: Weighting factor 5
• Average: Weighting factor 10
• Complex: Weighting factor 15

The *unadjusted use case weights (uUCW)* is calculated counting the number of use cases in each category, multiplying each category of use case with its weight and adding the products. The UAW is added to the UUCW to get the *unadjusted use case points* (*UUPC*).

3. The use case points are adjusted based on the values assigned to a number of technical and environmental factors.

Each factor is assigned a value between 0 and 5 depending on its assumed influence. A rating of 0 means the factor is irrelevant for the project; 5 means it is "essential".

The Technical Factor (TCF) is calculated multiplying the value of each factor by its weight and then adding all these numbers to get the sum called the *TFactor*. Finally, the following formula is applied:

TCF = 0.6 + (.01 * TFactor)

The Environmental Factor (EF) is calculated accordingly by multiplying the value of each factor by its weight and adding all the products to get the sum called the *Efactor*. The formula below is applied:

EF = 1.4 + (-0.03 * EFactor)

The *adjusted use case points (UCP*) are calculated as follows:

UCP = uUCP * TCF * EF

Karner [7] proposed a factor of 20 staff hours per use case point for a project estimate, while Sparks states that field experience has shown that effort can range from 15 to 30 hours per use case point [6]. Schneider and Winters recommend that the environmental factors should determine the number of staff hours per use case point [4].

According to our own studies and experience [1] [2] [10] [11] [12] [13] we agree with Sparks' point of view.

Example of use case points method application

Here we mention a project at COMPSIS Company located in Brazil [8][9]. The project's aim was to construct a software system used to register the data about customers of electronic toll collection. It was called TURS (Toll User Registration System).

The list of actors and identified use cases for the system are presented in Tables 1 and 2 with their respective complexities.

Table 1: Example of "real world" application of use case points (actors)

Actors	Complexity
Lane System	Average
Operational Manager	Complex
Vendor	Complex

Table 2: Example of "real world" application of use case points (use cases)

Use cases	Complexity
Pass registration	Average
Customer registration	Average
Order registration	Average
Order payment	Simple
Pass delivery	Simple
Interface with lanes	Simple
Customer list report	Simple
Financial report	Simple

The complexity for each actor and use case identified during this phase was determined following Karner's specification [7] and its criteria are synthesized in Tables 3 and 4.

Table 3: Criteria for actor complexity determination

Complexity	Definition	Weight
Simple	Represents a system with API available	1
Average	Represents an interaction with another system through a protocol There is a human interaction with a line terminal	2
Complex	Represents an interaction with a graphical user interface	3

Table 4: Criteria for use case complexity determination

Complexity	Definition	Weight
Simple	3 or less transactions	5
Average	Between 4 and 7 transactions	10
Complex	More than 7 transactions	15

Using the following formula:

$$uUCP = \sum_{i=1}^{6} n_i * W_i$$

The unadjusted use case weights (UUCW) were calculated, where:

uUCP => total of unadjusted UCP
i =1..6 => the list of complexity levels for actors and use cases.
n_i => total of actors and use cases identified and organized by complexity level
W_i => influence factor according to the complexity of factor and use case

In the case of TURS project, the calculation was:

uUCPActors = 1 * 2 + 2 * 3 = 8
uUCPUse Case = 5 * 5 + 4 * 10 + 2 * 15 = 95
uUCP = uUCPActors + uUCPUse Case = 103

The application of the Technical Factor (TCF) and Environmental Factor (EF) was not reported in this example by authors [8][9]

According to our experience (several e-government systems development [10] [11] [12]) “use case points” method has clear advantages compared to “line of code”, “function points”, “COCOMO” and other “traditional methods” in the effort evaluation field.

Currently we (the authors of this paper) are facing the effort evaluation for a Health Care Provider System including Technical Factor (TCF) and Environmental Factor (EF) whose results will be reported in the near future.

3. The Project Management Body of Knowledge use in software projects

The PMBOK presents Project Management practices in logical dimensional groups. One dimension describes "knowledge areas" while the other dimension describes project management processes split into five process groups

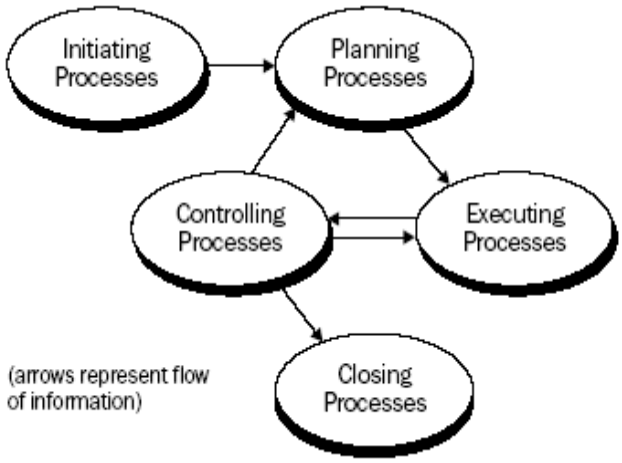
The PMBOK knowledge areas are:

i. Project Integration Management
ii. Project Scope Management
iii. Project Time Management
iv. Project Cost Management
v. Project Quality Management
vi. Project Human Resource Management
vii. Project Communications Management
viii. Project Risk Management
ix. Project Procurement Management

The 39 processes are organized into five process groups:

- Initiating Processes
- Planning Processes
- Executing Processes
- Controlling Processes
- Closing Processes

We can see theses five process groups:



We can mention, as PMBOK output (Some of the main PMBOK deliverables):

- Project Plan (with supporting details)
- Work Results and Change Requests
- Corrective Actions and Lessons Learned
- Project Charter (with constraints and assumptions)

The Project Plan implies to be:

- Created in Project Plan Development process in the early phases of a project lifecycle
- Updated throughout the project lifecycle
- Outlined as follows:
    - Project Charter
    - Project Management Approach or Strategy
    - Scope Statement
        - Project objectives
        - Project deliverables
    - Work Breakdown Structure (WBS)
    - Cost Estimates, Schedule and Responsibility Assignments for Deliverables
    - Measurement Baselines for Scope, Schedule and Cost
    - Major Milestones and Target Dates
    - Required Staff
    - Other Plans

Throughout the development of this paper we have strongly recommended [13] the use of PMBOK instead of "commercial" or "proprietary" methodologies for project management. We are suggesting the use of "use case points" method to support some of PMBOK processes/ knowledge areas.

4. "Use case points" method supporting PMBOK areas of knowledge

As pointed previously, we are applying "use case points" method in the context of PMBOK methodology (Health Care Provider System development).

The following table shows the use of "use case points" method considering PMBOK areas of knowledge

PMBOK areas of knowledge	Use case points method application
Project Integration Management	
Project Scope Management	Scope changes are "the rule" in software projects. Scope change quantification is an important application of use case points
Project Time Management	Programming (project calendar) is a very important area of use case points application.
Project Cost Management	Budgeting is a critical area of use case points application to effort estimation
Project Quality Management	Quantitative Software Quality Assurance planning topics could be an application of use case points
Project Human Resource Management	Human Resource allocation is a another critical area of use case points application to effort estimation
Project Communications Management	
Project Risk Management	Quantitative Risk Management planning topics could be an application of use case points
Project Procurement Management	Quantitative Project Procurement Management planning topics is an important area of application of use case points method

5. Conclusions

a. The use case points method is, perhaps, the best available tool for effort estimation in the context of software projects. It has been proposed to estimate the software development in early phases of software project and used by a lot of software organizations. Intuitively, UCP is measured by counting the number of actors and transactions included in use case models. Several tools to support calculating UCP have been developed. The acceptance of UCP as a generalized standard will be an important contribution to software industry.
b. PMBOK, used in software projects, is considered to have clear advantages regarding commercial methodologies. An aggressive commercial pressure, has evidently well positioned some of these commercial methodologies in the market. Therefore, the increasing adoption of PMBOK, in software development teams, will also be an important contribution to software industry.
c. Several PMBOK processes / knowledge areas can be supported applying “use case points” method.
d. Software development teams must use their own statistical data comparing programming / budgeting estimations against project execution data. This comparative activity will include teams into a permanent improvement process. If software development teams share these statistical data, in the future, software industry will use very consistent and useful tables when programming and budgeting tasks.

6. References

[1] R. Uzal, G. Montejano, D. Riesco, et al, “Software Engineers First Duty: the Preliminary Financial Feasibility Analysis of Real World Software Projects”, 5th IEEE International Symposium on Signal Processing and Information Technology (IEEE ISSPIT-2005), December 18-21, 2005, Athens, Greece

[2] N. C. Debnath, R. Uzal, G. Montejano, D. Riesco, “Software Projects Leadership: Elements to Redefine Risk Management" Scope and Meaning”, IEEE-EIT06. 6th IEEE International Conference on Electro/Information Technology (IEEE EIT-2006) May 7-10, 2006 Michigan State University East Lansing – Michigan – USA.

[3] www.pmi.org Project Management Institute, A guide to the project management body of knowledge (PMBoK® Guide) (Newtown Square, PA: Project Management Institute, 2004).

[4] Schneider, G. and Winters, J. *Applying Use Cases – A Practical Guide*. Addison-Wesley. 1998.

[5] Smith, J. The Estimation of Effort Based on Use Cases. Rational Software, White paper. 1999.

[6] Sparks, S. and Kaspcynski, K. The Art of Sizing Projects, Sun World. 1999.

[7] (http://www.sunworld.com/sunworldonline/swol-12-1999/swol-12-itarchitect.html).

[8] G. Karner, Resource estimation for Objectory projects, *Objective Systems SF AB*, 1993.

[9] T. C. Chiossi, and F. S. R. Germano, Software project management (Campinas, Brazil: Computing Institute – UNICAMP, 2001).

[10] T. C. Chiossi et al, ”The integration of use case points, COCOMO II and earned value analysis within PMBOK guide, International Conference on Software Engineering and Applications, IAESTED, 2006.

[11] R. Uzal, G. Montejano, D. Riesco, et al, "Designing a particular Balanced Scorecard to Manage the last and very difficult stage of a very important hydroelectric project", Proceeding of the International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications, The International Society for Computers and Their Applications, Rio de Janeiro, Brazil, pg 23-26, June 2003, ISBN 0-9742059-0-7.

[12] R. Uzal, G. Montejano, D. Riesco, M. Peralta, C. Salgado, "Improving the Management of the last stage of the Yacyreta Hydroelectric Project trhough the use of Kaplan and Norton's Balanced Scorecard Scheme", Proceedings of the 15th International Conference, Computer Applications in Industry and Engineering, the International Society for Computers and their Applications - ISCA, pg 252-255, San Diego, California, USA, November 7-9, 2002. ISBN: 1-880843-45-5. This publication is abstracted and indexed in INSPEC.

[13] J. Dujmovic, E. Petrolo, R. Uzal, A. Dasso, A. Funes, D. Riesco, G. Montejano, "Software Quality Assurance in a Project Based on Rapid Evolutionary Prototyping Methodology", 17th Anual International Conference. Proceedings of Computer Science. The Association of Management and the International Association of Management, Volume 17, number 3. Part C. August 6-8, 1999, San Diego, California, USA. ISGN:0-9668650-6-5. Pg 147-152. http://cs.sfsu.edu/People/jozo/respub.html

[14] Roberto Uzal, Germán Montejano, Daniel Riesco, “Pautas para optimizar la gestión de proyectos de software”, CACIC06, San Luis, del 24 al 27 de Octubre de 2006. ISBN CD: 950-609-050-5 ISBN impreso: 950-609-049-1

Transformación de Actividades SPEM por Medio de su Transformación en Relations a subProcesos BPMN

Fabio Zorzan
Departamento de Computación
Universidad Nacional de Río Cuarto,
Ruta 36 Km 601, 5800, Río Cuarto, Córdoba, Argentina
fzorzan@dc.exa.unrc.edu.ar

y

Daniel Riesco
Departamento de Informática
Universidad Nacional de San Luís,
Ejercito de los Andes 950, 5700, San Luís, Argentina
driesco@unsl.edu.ar

Resumen

El modelado de los procesos de negocio es de vital importancia en el desarrollo de toda industria, en particular, en la industria del software. Una forma de optimizar la producción es mediante la automatización de los procesos de negocio. Este trabajo aporta a la automatización total o parcial de la gestión de las actividades de los procesos de desarrollo de software, especificados con el Software Process Engineering Metamodel(SPEM), mediante workflows. Para lograr esta automatización, se transforman las actividades del proceso de desarrollo de software especificadas en SPEM en una especificación de subprocesos de la Business Process Modeling Notation (BPMN) definido por la Object Management Group (OMG).Esta transformación se realiza por medio del lenguaje Relations que forma parte de Query/Views/Transformations (QVT). La definición de subprocesos BPMN obtenida, puede ser transformada en una especificación en un lenguaje estándar para la implementación de procesos workflow, como ser Business Process Execution Language for Web Services (BPEL4WS) o XML Process Definition Languaje (XPDL). Así, de esta manera, la especificación en el lenguaje de implementación elegido será la entrada a cualquier motor Workflow. De esta forma es posible automatizar total o parcialmente la gestión de actividades de procesos de desarrollo de software especificados bajo SPEM.

Palabras claves: Workflow, SPEM, BPMN, QVT, Relations.

1 INTRODUCCIÓN

Los procesos de negocio son un conjunto de tareas lógicamente relacionadas, ejecutadas para obtener un resultado de negocio.
Los procesos de negocio pueden ser controlados y administrados por un sistema basado en software. Los procesos de negocio automatizados de esta manera se denominan workflow. Esta automatización resulta en una importante potenciación de las virtudes de dicho proceso. Se obtienen mejoras en cuanto a rendimiento, eficiencia y productividad de la organización.
El paradigma workflow ofrece interoperabilidad con otros sistemas, ejecución en ambientes distribuidos, facilidades para el monitoreo y manejo de recursos humanos[1].
El caso de la industria del desarrollo de software, no es diferente al del resto de las industrias. Dentro de ella, se encuentran los procesos de negocios tendientes a la construcción o generación de un producto (software) de calidad en un tiempo determinado[2]. El proceso de negocio mas importante dentro de la industria de desarrollo de software es conocido como "metodologías de desarrollo", encargadas de guiar la producción. Actualmente, los ingenieros de software trabajan para optimizar los procesos de desarrollo. Los desarrolladores de las herramientas de ingeniería de software pueden explotar la conexión entre la administración de proceso de desarrollo de software y workflow[3].
Este trabajo aporta a la optimización del proceso de producción de software mediante la automatización total o parcial de la gestión de actividades de las metodologías de desarrollo de software especificados en SPEM. Para lograr esta automatización, se propone una traducción de las actividades del proceso de desarrollo de software especificado en SPEM[4] a una especificación de sub-procesos Workflow basado en el estándar BPMN[5] aceptado por la OMG. Esta traducción se obtiene a través de una transformación definida mediante el lenguaje Relations que forma parte de QVT[6]. La transformación está definida entre el Metamodelo SPEM y el metamodelo BPMN. La hipótesis de trabajo es plantear al proceso de desarrollo de software como un tipo proceso de negocio particular automatizándolos en todo o en parte a través de un motor de workflow.
Es importante a la hora de dar claridad a los procesos de negocio, la descomposición de estos en subprocesos de alto nivel. El objetivo es transformar las actividades del proceso de desarrollo de software en sub-procesos de un workflow para poder lograr la automatización total o parcial de su gestión.
El trabajo está organizado de la siguiente manera. En la sección 2 se presenta al SPEM con sus características. La sección 3 presenta la tecnología workflows y la notación BPMN. En la sección 4 se presenta el lenguaje de transformación Relations, parte de QVT. La sección 5 describe la transformación del metamodelo SPEM al metamodelo BPMN. Por ultimo, en la sección 6 están las conclusiones.

2 SPEM

Los procesos en el desarrollo de software pueden ser vistos como productos, ya que están constantemente cambiando y evolucionando. También deben ser administrados y configurados para adaptarlos a las organizaciones y a las nuevas necesidades del entorno, agregando de esta forma la necesidad de un estándar unificado en esta área, esto debido a que cada una de estas técnicas y procesos definió sus propios estándares y terminologías usando incluso diferentes significados para la misma palabra.

Para especificar las actividades propuestas por un proceso de desarrollo particular y de esta forma proveer una solución a la necesidad antes planteada, la OMG definió un metamodelo para la Ingeniería de Procesos de Software (SPEM).
Para la definición de nuevos Lenguajes la OMG define una arquitectura basada en cuatro niveles de abstracción que van a permitir distinguir entre los distintos niveles conceptuales que intervienen en el modelado de un sistema. Esos niveles se denominan M0, M1, M2 y M3 y se describen a continuación:

- **El nivel M0 – Las instancias.** El nivel M0 modela el sistema real, y sus elementos son las *instancias* que componen dicho sistema. Un elemento de este nivel es por ejemplo el cliente llamado Pablo Gonzalez.
- **El nivel M1 – El modelo del sistema.** Los elementos del nivel M1 son los *modelos* de los sistemas concretos. Existe una relación muy estrecha entre los niveles M0 y M1, los conceptos del nivel M1 definen las *clasificaciones* de los elementos del nivel M0. En este nivel esta definido el concepto cliente.
- **El nivel M2 – El modelo del modelo (el *metamodel*o**). Los elementos del nivel M2 son los lenguajes de modelado. El nivel M2 define los elementos que intervienen a la hora de definir un modelo del nivel M1. Aquí también existe una gran relación entre los conceptos de los niveles M1 y M2 donde los elementos del nivel superior definen las *clases* de elementos válidos en un determinado modelo de nivel M1. Conceptos de este nivel son Clase, Atributo, etc.
- **El nivel M3 – El modelo de M2 (el *meta-metamodel*o**). Finalmente, el nivel M3 define los elementos que constituyen los distintos lenguajes de modelado. La OMG ha definido un lenguaje para describir los elementos del M3 llamado Meta-Object Facility (MOF) [7], un elemento de este nivel en el clasificador, el concepto de clase definido en M2 es un clasificador.

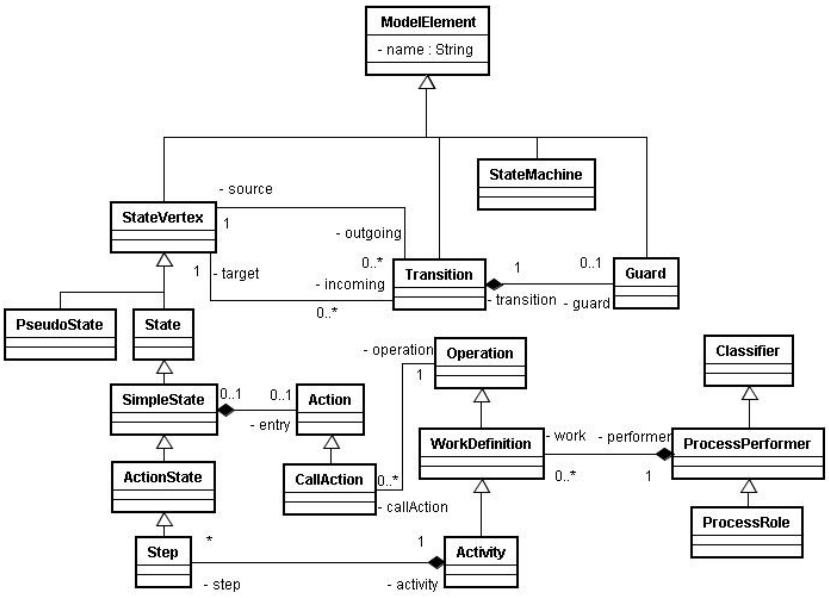


Fig. 1. Vista del metamodelo SPEM con las metaclases involucradas en este trabajo.

SPEM describe un metamodelo genérico para la descripción de procesos software concretos que está basado en MOF y utiliza UML como notación de modelado. Por tanto, se basa en los principios de orientación a objetos.

El metamodelo SPEM sirve como plantilla para la creación de modelos de procesos concretos, como podrían ser el "Proceso Unificado de Desarrollo de software de Rational" (RUP) o el modelo de evaluación y mejora de procesos de ISO 15504. Por tanto, SPEM es un metamodelo del nivel M2 de MOF, mientras que estos procesos citados se definirían en base a SPEM en el nivel M1.
Este trabajo se centra en el paquete estructura de procesos de SPEM que define los elementos estructurales principales en la construcción de la descripción de un proceso y el paquete maquina de estados que define principalmente las transiciones entre las "actividades" del proceso. La Figura 1 muestra una vista del metamodelo SPEM con las metaclases involucradas en la transformación.

3 WORKFLOW

Un workflow se define como la automatización total o parcial de un proceso de negocio, durante la cual documentos, información o tareas son intercambiadas entre los participantes conforme a un conjunto de reglas procedimentales preestablecidas [8].
Un workflow comprende un número de pasos lógicos, conocidos como actividades. Una actividad puede involucrar la interacción manual o automática con el usuario.
Un motor workflow es un sistema de software que controla la ejecución de las actividades definidas en el workflow. La WfMC ha definido un Modelo de Referencia Workflow (Workflow Reference Model). Este modelo define 5 interfaces para la interoperabilidad de diferentes productos con un motor workflow.
En este trabajo interesa la interfaz 1 que especifica el formato de intercambio común para soportar la transferencia de definiciones de procesos entre productos diferentes, utilizando un lenguaje de definición de procesos como el XML Process Definition Language – (XPDL)[9] definido por la WfMC o el Business Process Execution Language for Web Services(BPEL4WS)[10] adoptado por OASIS. XPDL permite escribir especificaciones de procesos workflow de manera estandarizada. Esto significa que cualquier definición de proceso que cumpla con todos los requisitos establecidos en la interfaz 1 podrá ser tomada como entrada por cualquier motor workflow que respete el estándar establecido por la WfMC, por ejemplo OFBiz Workflow Engine [11] o Open Business Engine [12].
BPEL4WS es un lenguaje para la especificación de procesos de negocio, el cual permite especificar procesos de negocio basados en servicios Web, esto es, que sólo pueden importar y exportar funcionalidad mediante servicios Web. La especificación inicial (BPEL4WS 1.0) fue desarrollada por IBM, Microsoft y BEA . WebSphere Process Server de IBM[13] y BPEL Process Manager[14] de Oracle son ejemplos de motores de workflow que implementan BPEL4WS.
Es importante a la hora de modelar un proceso de negocio poder utilizar una herramienta independiente de la implementación, así, de esta manera, poder utilizar la especificación del proceso de negocio para diferentes plataformas. Una herramienta de estas características que esta siendo muy utilizada por grandes empresas es BPMN.

3.1 BPMN

La OMG junto con la Bussines Process Modeling Initiative(BPMI) han desarrollado la notación BPMN para el modelado de procesos de negocio. BPMN define una notación para la definición de procesos de negocio, lo que es una plataforma independiente con respecto a definiciones específicas de procesos de negocio, por ejemplo XML Process Definition Language (XPDL)[9] o Business Process Execution Language for Web Services (BPEL4WS)[10]. Esta notación define una representación abstracta para la especificación de procesos ejecutables de negocio que se ejecutan dentro de una empresa (con o sin intervención humana); y puede colaborar con otro proceso de

negocio independiente ejecutado en otra unidad de negocio o empresa. Partiendo de un modelo especificado en BPMN se puede obtener, mediante un mapping, la definición de un proceso de negocio en un lenguaje especifico como ser XPDL o BPEL4WS. En [5] esta definido el mapping de BPMN a BPEL4WS.
Los elementos de la notación se pueden clasificar en elementos de flujo, de conexión, swinlanes y artefactos. Estos elementos que forman parte de la notación están especificados en el metamodelo BPMN[15]. Este metamodelo esta definido en el nivel M2 de la OMG y esta basado en MOF.

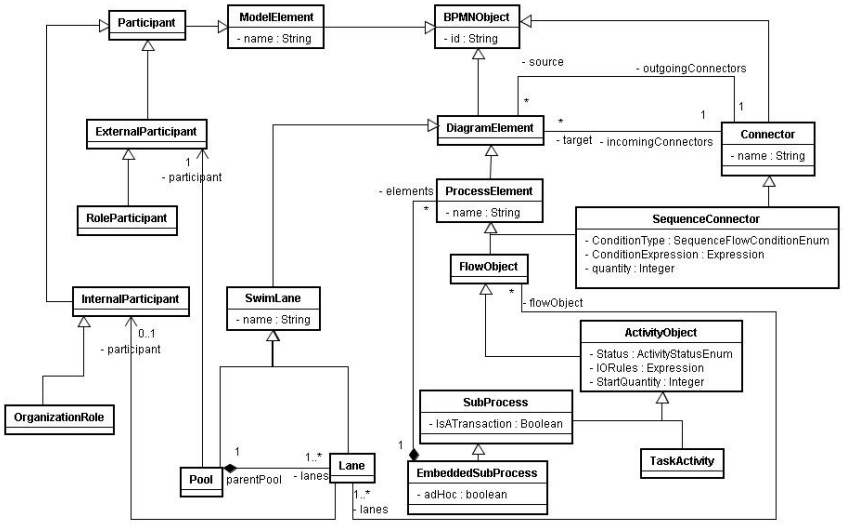


Fig. 2. Vista del metamodelo BPMN con las metaclases involucradas en este trabajo

La figura 2 muestra una vista del metamodelo BPMN donde se pueden ver las metaclases del metamodelo BPMN involucradas en la relación presentada en este trabajo. La metaclase *EmbeddedSubProcess* representa los subprocesos de alto nivel que contienen tareas(*TaskActivity*), conectores (*Connector*), etc.
La metaclase *EmbeddedSubProcess* representa un subproceso dentro de un proceso de negocio. *EmbeddedSubProcess*, al heredar de *DiagramElement*, posee un conjunto de conectores representados por la metaclase *Connector*. *Connector* representa las conexión entre los elementos de un diagrama BPMN. *SequenceConnector* es una subclase de *Connector*. *SequenceConnector* está involucrada en la relación y representan el orden en que las actividades se ejecutan.

4 QVT

El planteamiento QVT[6] se basa principalmente en: la definición de un lenguaje para las consultas (Queries) sobre los modelos MOF, la búsqueda de un estándar para generar vistas (Views) que revelen aspectos específicos de los sistemas modelados, y finalmente, la definición de un lenguaje para la descripción de transformaciones (Transformations) de modelos MOF.

4.1 Relations

En este trabajo se presentará el componente de QVT que tiene como objetivo definir transformaciones, en especial, el lenguaje declarativo relations.
El lenguaje relations es una especificación declarativa de relaciones entre metamodelos MOF. Este lenguaje permite realizar pattern matching de objetos complejos y definir templates de creación de

objetos. El trace de los elementos de los modelos involucrados en las transformaciones son creados explícitamente.
Estas transformaciones describen relaciones entre un meta-modelo fuente F y un meta-modelo objetivo O, ambos metamodelos deben estar especificados en MOF. Luego esta transformación definida se utiliza para obtener un modelo objetivo que es una instancia del metamodelo O a partir de un modelo fuente que es una instancia del metamodelo F. Una característica muy importante de estas transformaciones es que pueden ser bidireccionales (multidimencionales también).
Una transformación especifica un conjunto de relaciones que deben cumplir los elementos de los modelos involucrados. Una relación especifica una relación entre elementos de los modelos candidatos y consiste de dos o mas dominios, y dos restricciones denominadas cláusula **guard** (o cláusula **when**) y cláusula **where**. Cada dominio define a un modelo de candidato. Cada dominio tiene patrones. Un patrón puede ser considerado un template para los objetos y sus propiedades, que deben ser encontrados, modificados o creados en los modelos involucrados para satisfacer la relación entre los modelos.

5 TRANSFORMACIÓN DEL METAMODELO SPEM AL METAMODELO BPMN

En este punto se presenta la transformación del metamodelo SPEM al metamodelo BPMN, primero dando un contexto general de las componentes intervinientes en la transformación y luego se muestra, en lenguaje Relacions de QVT, la definición de la transformación.

5.1 Contexto General de la Transformación

La vista general de la transformación de procesos de desarrollo de software basados en SPEM a workflows puede ser particionada en tres niveles: Metamodelo, Definición/Modelo y Ejecución, como lo muestra la figura 3.

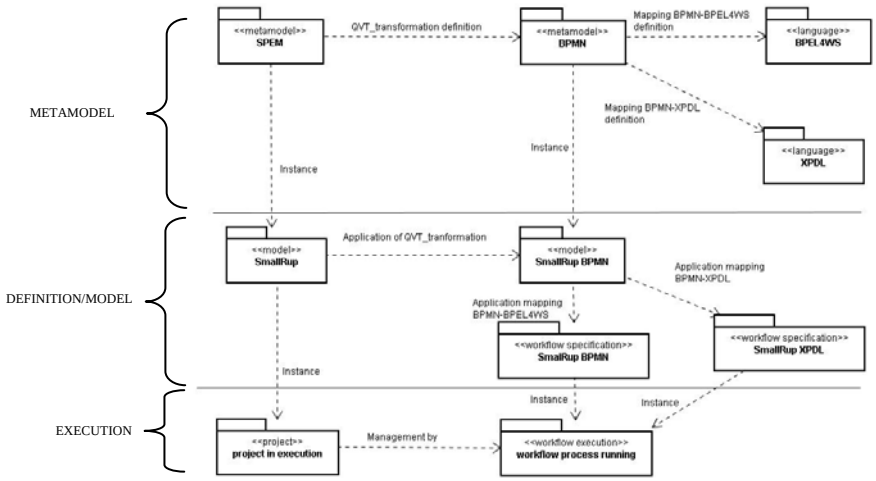


Fig. 3. Vista general de la transformación del metamodelo SPEM al metamodelo BPMN.

A nivel metamodelo se encuentran los metamodelos objetivos de la transformación definida en este trabajo, el metamodelo SPEM y el metamodelo BPMN, entre los cuales esta definida la transformación mediante el lenguaje QVT. A su vez en este nivel se encuentran las definiciones de los mapping entre el metamodelo BPMN y los diferentes lenguajes de implementación de workflow, en este caso los lenguajes XPDL y BPEL4WS.

Pasando al nivel de modelo/definición se encuentran los modelos específicos que definen un proceso de desarrollo de software concreto, como por ejemplo SmallRUP[16], y a partir de éste, por aplicación de la transformación definida en QVT a nivel metamodelo, se obtiene el modelo BPMN que define a SmallRUP como un proceso de negocio. También en este nivel se encuentran la aplicación de los mapping entre el metamodelo BPMN y los diferentes lenguajes de definición de procesos, que como resultado de la aplicación de estos mapping se obtiene la definición de SmallRUP en un lenguaje(XPDL o BPEL4WS). Ésta definición se utiliza como entrada para la definición de procesos en un motor Workflow que implemente el lenguaje.
Por último, en el nivel de ejecución, se encuentran los proyectos de desarrollo de software que siguen como metodología de desarrollo de software a SmallRUP y que son administrados automáticamente a través de motores de workflow que siguen como especificación de procesos de negocio a la definida en el nivel anterior.

5.2 Transformación del Metamodelo SPEM al Metamodelo BPMN

Este trabajo hace una contribución a la mejora en la gestión de las actividades de los procesos de desarrollo de software que están basados en el estándar SPEM. Para esto se propone la transformación de las actividades de los procesos software basado en SPEM a subprocesos BPMN que pueden ser automatizados, previa transformación a XPDL o BPEL4WS, mediante la utilización de un workflow estándar. De esta forma se puede utilizar una herramienta workflow que implemente el estándar de la WfMC(XPDL) o el estándar de OASIS(BPEL4WS) para asistir en la gestión de los procesos de desarrollo de software.
Para poder especificar las reglas de transformación se definió una correspondencia entre metaclases del metamodelo SPEM y metaclases del Metamodelo BPMN.
Para la definición de las reglas de transformación de metamodelos se adoptó el lenguaje Relations de QVT. QVT permite hacer transformación de diferentes formas (unidireccionales, bidireccionales, solamente chequeo de correspondencias entre modelos, entre otras). En este caso la transformación solo necesita ser definida en la dirección Metamodelo SPEM hacia Metamodelo BPMN. De esta manera los elementos del metamodelo SPEM están marcados como checkonly y los elemento del metamodelo BPMN están marcados como enforced, para que de esta forma la ejecución de la transformación cree los elementos del modelo BPMN que se corresponden a los elementos del modelo fuente especificado en SPEM.

5.2.1 Definición de la Transformación de Actividades SPEM a SubProcessos BPMN

En esta sección se presenta relación definida para la transformación de actividades SPEM a SubProcesos BPMN. La transformación entre el metamodelo SPEM y el Metamodelo BPMN se define de la siguiente manera:
Transformation SpemWorkflow(spem: Spem, bpmn: BPMN)
Esta transformación toma un modelo spem que es una instancia del metamodelo SPEM y un modelo bpmn que es una instancia del metamodelo BPMN.
Para una mejor compresión de la explicación de la definición de la relación presentada que forman parte de la transformación QVT, los nombres de las metaclases serán escritos con fuente de letra cursiva.
Debajo se muestra la especificación de la relación **activitySPEMToSubprocessBPMN** que define la correspondencia entre la metaclase *State*, (en particular para los objetos de la metaclase *State* que tienen definida la relación *entry* con un objeto de la metaclase *CallAction* y a su vez, el objeto *CallAction* debe estar relacionada un objeto de la metaclase *Activity* del metamodelo SPEM) y la

metaclase *EmbeddedSubProcess* del metamodelo BPMN. Las transiciones (metaclase *Transition* de SPEM) de entrada y salida del *ActionState*, representadas por los atributos *outgoing* e *incoming* (atributos heredados de la metaclases *StateVertex*) se transforman en conectores de secuencia (*SequenceConnector*) de entrada y salida respectivamente de la metaclase *EmbeddedSubProcess* del metamodelo BPMN. Esta transformación está definida por dos template de creación de objetos, uno para los conectores de entrada y otro para los de salida. Dentro de los templates de los conectores se define el atributo *conditionExpression* que representa la condición bajo la cual se puede ejecutar la transición definida por el conector. La expresión del conector se obtiene a través de la guarda de la transición de SPEM que se corresponde al conector. La correspondencia entre la guarda de SPEM y la expresión de BPMN esta definida en la cláusula **where** de la relación. La cláusula **where** obliga a cumplir la relación *guardToExpression*, entre la guarda de las transiciones y la expresión de los conectores. También dentro de la cláusula **where** de obliga al cumplimiento de la relación stepSPEMToTaskBPMN que define la correspondencia entre un *Step* de SPEM y una *Task* BPMN.

```
top relation activitySPEMToSubprocessBPMN
{
 nameActivity : String;
 nameTransitionIn : String;
 nameTransitionOut : String;
 nameRole : String;

ckeckonly domain spem actionState: ActionState
      {
      entry = callAction : CallAction
       {
        operation = activity : Activity
                  {
                   name = nameActivity,
                   peformer = performe : ProcessRole{name = nameRole},
                   assistente = assistente : ProcessRole {}
                  }
       }
      incoming = transitionInSpem : Transition
       {
        guard = guardTransitionIn : Guard{},
        name = nameTransitionIn
       },
      outgoing = transicionOutSpem : Transition
       {
        guard = guardTransitionOut : Guard{},
        name = nameTransitionOut
       }
} //fin ckeckonly domain spem actionState: ActionState
```

```
enforce domain bpmn subprocess: SubProcess
        {
// El id se forma concatenando el string "id" con el nombre de la actividad.
        id = "id"+ nameActivity,
        name = nameActivity,

// Para definir el template para los conectores BPMN de entrada se utiliza la
// información de las  transiciones de entrada al step de SPEM.
    incomingConnectors = sequenceConnectorInBPMN : SequenceConnector
        {
         id = "id"+nameTransitionIn,
         name = nameTransitionIn,
         conditionExpression = conditionExpressionIn : Expression{},
         quantity = 1
        },

// En la definición del temºplate para los conectores BPMN de salida se
// utiliza la información de las  transiciones de entrada al step de SPEM.
    outgoingConnectors = sequenceConnectorOutBPMN : SequenceConnector
        {
         id = "id"+nameTransitionOut,
         name = nameTransitionOut,
         conditionExpression = conditionExpressionOut : Expression{},
         quantity = 1
        },

// Los lanes son creados con la información del Performer que llevan a cabo
// la actividad de SPEM.
// Además en este template se crea el pool al que pertenece el lane, el pool se
// crea con un id  constante para asegurar la creación de un único pool en la
// transformación.
    lanes = lane : Lane{} // fin de Lane
    }; // fin de  domain bpmn subprocess: SubProcess
where
{
// Realiza el mapping entre los pasos de la actividad y las tareas del
// subProceso
stepSPEMToTaskBPMN(activity, subProcess);

// Obliga el cumplimiento  de la relación
// ProcessRoleToOrganizationRoleBPMN que define la correspondencia
// entre  un ProcessRole de SPEM y  OrganizationRole de BPMN.
    processRoleToOrganizationRoleBPMN(processRole, participantBPMN);

// Exige el cumplimiento  de la relación guardToExpression que define la
// correspondencia entre  un Guard de SPEM y  una Expession de BPMN.
guardToExpression(guardTransitionIn, condicionExpressionIn);
guardToExpression (guardTransitionOut, condicionExpressionOut);

//  Define la creación del lane a partir e la información del performe.
processRoleToLane(performe,lane);
}
```

6 CONCLUCIONES

Este trabajo se centra en hacer una contribución a la optimización de los procesos de desarrollo de software, considerando al proceso de desarrollo de software como un proceso de negocio particular, y con esto, pueden ser automatizados en todo o en parte a través de un motor de workflow. Para lograr esta automatización de los procesos de desarrollo de software, se definió una transformación de actividades SPEM a subprocesos de un workflow para poder lograr la automatización total o parcial de su gestión. Teniendo en cuenta esto, las actividades del proceso de desarrollo de software se transforman en una especificación de subprocesos BPMN y así poder utilizar un motor workflow que siga el estándar de la WfMC u Oasis (dependiendo del lenguaje de implementación elegido), y de esta forma, automatizar total o parcialmente la gestión de las actividades de los procesos de desarrollo de software especificados con el estándar de la OMG denominado SPEM.
La relación presentada en este trabajo representa una parte central de la transformación entre los metamodelos SPEM y BPMN, debido a que define la relación entre las actividades del proceso de desarrollo de software y los subprocesos BPMN.
Usar como objetivo de la transformación el metamodelo BPMN, tiene la ventaja que los modelos obtenidos son independientes de la plataforma de automatización, esto es, independiente del motor workflow que se utilice para la administración de los proyectos de desarrollo de software basados en alguna metodología especificada en SPEM.
Un caso práctico de la aplicación de la transformación, es su aplicación a la especificación en SPEM de las actividades del SmallRUP, esto da como resultado una especificación de subprocesos workflow en BPMN. Luego, aplicándole a la especificación BPMN el mapping a BPEL4WS (o XPDL), se obtiene una especificación de procesos que puede ser tomada por cualquier motor workflow que interprete una especificación BPEL4WS (o XPDL), y así, poder administrar automáticamente, por medio de un motor workflow, las actividades de los proyectos de desarrollo de software que utilicen como metodología de desarrollo a SmallRup.
El beneficio de esta automatización también se aprecia teniendo en cuenta el dinamismo de los cambios en los procesos de desarrollo de software, con lo cual, cualquier cambio en la especificación de las actividades de un proceso de desarrollo de software puede ser propagado a la especificación Workflow de dichas actividades y así adaptar rápidamente la especificación del workflow para la automatización de las actividades de los procesos de desarrollo de software.
Esta transformación optimiza la construcción del software debido a que se dispone de un sistema automatizado (motor workflow) que administrará los recursos y organizará a un equipo de ingenieros de software en el transcurso del desarrollo de un proyecto en particular. El proceso de desarrollo adopta todas las ventajas propias de un proceso de negocio.

REFERENCIAS

[1] Daniel K.C. Chan, Karl R.P.H. Leung, "Software Development as a Workflow Process," apsec, p. 282-291, Fourth Asia-Pacific Software Engineering and International Computer Science Conference (APSEC'97 / ICSC'97), IEEE 1997.

[2] N. Debnath, D. Riesco, G. Montejano, et al, “Supporting the SPEM with a UML Extended Workflow Metamodel”, ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'06). Conference to be held in Dubai/Sharjah during March 8-11, 2006, www.ieee.org .

[3] Anthony Barnes, Jonathan Gray, "COTS, Workflow, and Software Process Management: An Exploration of Software Engineering Tool Development" aswec, p. 221-232, 2000 Australian Software Engineering Conference, IEEE 2000.

[4] Object Management Group, “Software Process Engineering Metamodel Specification”; An Adopted Specification of the Object Management Group, Inc; Version1.1 formal/05-01-06; January 2005, http://www.omg.org/docs/formal/05-01-06.pdf, último acceso Marzo 2006.

[5] Object Management Group "Business Process Modeling Notation (BPMN) Specification". Final Adopted Specification dtc/06-02-01, http://www.bpmn.org/Documents/OMG Final Adopted BPMN 1-0 Spec 06-02-01.pdf, último acceso Mayo 2007.
[6] Object Management Group, "Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification" Final Adopted Specification ptc/05-11-01, http://www.omg.org/docs/ptc/05-11-01.pdf, último acceso Junio 2007
[7] Object Management Group "Meta Object Facility (MOF) Core Specification" OMG Available Specification. Version 2.0. formal/06-01-01, http://www.omg.org/docs/formal/06-01-01.pdf, último acceso Diciembre 2006.
[8] Rob Allen, Open Image Systems Inc., United Kingdom Chair, WfMC External Relations Committee; "The Workflow Handbook 2001"; Workflow Management Coalition; October 2001.
[9] Workflow Management Coalition, Workflow Standard – Workflow Process Definition Interface -XML Process Definition Language, Workflow Management Coalition , WfMC-TC-1025, 2002, http://www.wfmc.org/standards/docs/TC-025_10_xpdl_102502.pdf , último acceso Octubre 2006.
[10] BEA, IBM, Microsoft, SAP and Siebel, "Business Process Execution Language for Web Services Version 1.1" , S. Thatte, et al., May 2003, ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf, último acceso, Mayo 2007.
[11] OFBiz Workflow Engine, http://incubator.apache.org/ofbiz/docs/workflow.html, último acceso Abril 2007.
[12]Open Business Engine, http://obe.sourceforge.net/ , último acceso Marzo 2007.
[13] IBM, "WebSphere Process Server", http://www-306.ibm.com/software/integration/wps/, último acceso Junio 2007
[14] Oracle, "BPEL Process Manager" http://www.oracle.com/technology/products/ias/bpel/index.html, último acceso Abril 2007.
[15] Object Management Group, BPMN Documents "BPMNModel UML Documentation". Draft Specificaction, http://www.bpmn.org/Documents/BPMNMetaModel.zip , último acceso Octubre 2006.
[16] Gary Pollice "Using the RUP for small projects: Expanding upon Extreme Programming", A Rational Software White Paper – 04/08/15, ftp://ftp.software.ibm.com/software/rational/web/whitepapers/2003/tp183.pdf, último acceso Mayo 2007.

Um Gerenciador de Artefatos para Ambiente de Desenvolvimento Distribuído de Software

César Alberto da Silva
Universidade Estadual de Maringá (UEM), Departamento de Informática
Maringá, Paraná, Brasil
cesaralberto91@hotmail.com

e

Elisa Hatsue Moriya Huzita
Universidade Estadual de Maringá (UEM), Departamento de Informática
Maringá, Paraná, Brasil
elisa@din.uem.br

Abstract

The increasing complexity of the activities to be executed in a project of software development demands a greater interaction among the participants of the project, mainly they are dispersed geographically. The artefacts generated by execution of activities during the project development, consist of programs, models and documents that need to be stored. A greater interaction among the people, increase the necessity of exchange of information, artefacts, ideas among them. It also increase, the possibility to occur eventual conflicts when more than one person will be involved in an activity interacting one with other. So, mainly in a distributed software development environment it is import to make available an artefact manager. This paper presents an artefact manager. When is considered the scenario where the team members are in geographically distinct places, it is necessary to provide ways to treat the conflicts that would occur when more than one person try to update the artefacts concurrently. So this proposal intends to include one solution for the treatment of this situation, and so to maintain the artefacts version consistent.

Keywords: Artefact, Conflict, Awareness, ADDS.

Resumo

A crescente complexidade das atividades a serem realizadas em um projeto de desenvolvimento de software exige uma maior interação entre os participantes do projeto, principalmente se estas estiverem geograficamente dispersas. Os artefatos resultantes de atividades realizadas durante o desenvolvimento de um projeto, constituem-se de programas, modelos e documentos textuais que precisam ser armazenados. Com uma maior interação entre as pessoas, aumenta a necessidade de troca de informações, artefatos, idéias entre elas. Aumenta também, a possibilidade de eventuais conflitos quando mais do que uma pessoa estiver envolvida em uma atividade interagindo para gerar e, posteriormente, persistir o artefato resultante. Assim, é importante que esteja disponível, principalmente em um ambiente de desenvolvimento distribuído de software, um gerenciador de artefatos. Este trabalho têm por objetivo construir um gerenciador de artefatos. Em se tratando de um cenário em que os membros da equipe podem estar em locais geograficamente distintos, uma das peculiaridades desse gerenciador é o de incluir uma proposta de resolução para o tratamento de conflitos que podem surgir quando mais de uma pessoa estiver tentando realizar atualizações em artefatos.

Palavras-chaves: Artefato, Conflitos, Percepção, ADDS.

1 INTRODUÇÃO

Os projetos de desenvolvimento de software têm, progressivamente, aumentado de tamanho e complexidade, sendo cada vez mais comum sua realização por equipes de médio porte (entre dez e vinte desenvolvedores) e grande porte (acima de vinte desenvolvedores). Com as facilidades de comunicação proporcionadas pela Internet, a necessidade de experiência em diversas áreas de conhecimento e a pressão por cronogramas mais restritos, acabam fazendo com que alguns projetos sejam desenvolvidos por diversas equipes trabalhando cooperativamente. Além disso, as dificuldades em reunir os especialistas necessários em um mesmo local físico e a delegação do desenvolvimento de determinados componentes para outras empresas, são exemplos de fatores que podem exigir que as equipes participantes de um projeto estejam geograficamente distribuídas [12]. As organizações têm cada vez mais utilizado o desenvolvimento de software remoto como uma facilidade adicional, levando a o que é conhecido como Desenvolvimento Distribuído de Software (DDS) [7] [9].

No entanto, este novo cenário traz desafios para o processo de desenvolvimento. Por exemplo, o trabalho de equipes geograficamente distribuídas dificulta o controle de alterações nos componentes de um projeto em desenvolvimento [19]. Mesmo quando precedida de uma definição precisa das interfaces entre os componentes, a realização de um projeto pode exigir que diversos desenvolvedores alterem, simultaneamente, os mesmos componentes. Esta situação exige a adoção de políticas adequadas para manter a consistência entre os componentes, da versão atual, do projeto ou permitir que essa consistência seja posteriormente restituída.

O desenvolvimento do software deve tratar de duas fontes de complexidade: a complexidade do artefato que está sendo produzido e a complexidade das atividades em torno desse artefato. Modelos de processo de software tentam ajudar as equipes a lidar com a complexidade das atividades, enquanto técnicas focam no artefato [6].

O trabalho cooperativo e a utilização de técnicas de *awareness* ajudam a diminuir as dificuldades para a realização de atividades que são executadas por mais de uma pessoa. Desenvolvedores pertencentes a uma equipe de projeto podem estar dispersos geograficamente, apoiados por decisões estratégicas, motivados por redução de custos e aumento de produtividade. Assim, é desejável que as atividades sejam apoiadas por algum software (*groupware*), que permite minimizar o retrabalho, afim de manter a consistência e uniformidade dos artefatos produzidos.

Um artefato é resultado de uma atividade e pode ser utilizado posteriormente como matéria prima para aquela ou para outra atividade afim de gerar novos artefatos. Dessa forma, uma atividade pode consumir artefatos (de entrada) e gerar novos artefatos (de saída). Artefatos são freqüentemente persistentes, armazenados em repositórios, e possuem versões [18].

Assim, embora o gerenciamento das modificações efetuadas em artefatos seja um fator importante quando estes são compartilhados por várias pessoas em um ambiente de desenvolvimento distribuído de software, as soluções apresentadas na literatura, conforme serão vistas na seção 4, tratam principalmente de resolução de conflitos de artefatos no formato código fonte, deixando uma lacuna no que se refere a artefatos no formato XMI [17].

O objetivo deste trabalho é apresentar uma solução para os problemas, relacionados ao armazenamento de artefatos no repositório, que poderão ocorrer na execução das atividades de um projeto.

O gerenciador de artefatos ora apresentado, está situado, em termos da arquitetura de um ambiente de desenvolvimento, entre o usuário e o repositório de artefatos. O repositório de artefatos é responsável por manter e controlar as versões dos artefatos, podendo ser, neste caso utilizado, por exemplo, o SubVersion[1]. O gerenciador de artefatos adiciona novas políticas de acesso ao repositório de artefatos, facilita a interação dos desenvolvedores com o repositório de artefatos, mantém os desenvolvedores atualizados sobre o andamento das atividades do projeto e permite resolver conflitos de artefatos no formato XMI, em particular os artefatos gerados pela ferramenta Requisite [1].

[1]Version Control System. Disponível em: <http://subversion.tigris.org>.

Este artigo está organizado como segue: nas seções 2 e 3 serão apresentados, respectivamente, os conceitos de desenvolvimento distribuído de software e repositório de artefatos, na seção 4 serão apresentados os trabalhos relacionados, na seção 5 será apresentada o gerenciador de artefatos, o contexto para validação, bem como a forma dos experimentos do gerenciador de artefatos. Finalmente, na seção 6 são apresentadas as conclusões e trabalhos futuros.

2 DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

Ao longo dos anos o software se tornou um componente vital da maioria dos negócios. O sucesso das organizações depende da utilização do software como uma arma competitiva. Na década de 80, muitas organizações começaram a experimentar o desenvolvimento de software remoto como uma facilidade a mais [7].

A crescente globalização do ambiente de negócios e da economia tem afetado, diretamente, o mercado de desenvolvimento de software. Os engenheiros de software vêm reconhecendo, há algum tempo, a profunda influência do desenvolvimento global de software na globalização dos negócios e, através de reações alarmistas estão se movimentando para encontrar um modelo de negócio que possa atender a este mercado. Recentemente, a atenção está se voltando para o entendimento dos fatores que permitem às multinacionais e às corporações virtuais a operar com sucesso ultrapassando as fronteiras, geográficas e culturais, em busca de vantagens competitivas como baixos custos, maior produtividade e melhor qualidade na área de desenvolvimento de software [5] [7].

Este fenômeno é alimentado por fatores tais como: o acesso a uma grande quantidade de mão de obra especializada, redução nos custos do desenvolvimento, presença global e proximidade ao consumidor. Apesar do sucesso de várias equipes globais, as pesquisas revelam que a distância contribui para aumentar a complexidade nos processos organizacionais. Primeiramente, os processos de comunicação e coordenação são afetados pela distância, com consequências diretas na definição, construção, testes e entrega do software ao cliente final, assim como no gerenciamento do desenvolvimento [11].

Devido à necessidade de se manter equipes geograficamente dispersas, várias ferramentas e ambientes têm sido construídos para ajudar no controle e coordenação dessas equipes. O desenvolvimento distribuído de software (DDS) tem sido caracterizado principalmente pela colaboração e cooperação entre equipes que realizam atividades em conjunto, mas estão localizados temporal e fisicamente distantes, acrescentando assim, novos desafios ao processo de desenvolvimento de software [4] [20]. Em particular, o gerenciamento dos artefatos produzidos na execução das atividades do projeto, demanda uma maior atenção por parte do gerente de projeto [11].

3 REPOSITÓRIO DE ARTEFATOS

O repositório de artefatos é utilizado para o controle de versões dos artefatos que são gerados pelos responsáveis pela execução de atividades de um projeto. Os mais utilizados são: SubVersion, CVS[2], ClearCase[3] e SourceSafe[4].

Em um ambiente distribuído de desenvolvimento de software é desejável que as atividades de projeto sejam executadas de forma cooperativa. O acesso ao repositório de artefatos pode ser realizado por qualquer membro da equipe que tenha recebido atribuição para execução de uma atividade em um determinado projeto.

No entanto, mesmo as atividades sendo executadas cooperativamente, existem momentos aonde os desenvolvedores necessitam trabalhar individualmente. Portanto, é interessante que cada desenvolvedor tenha uma cópia do artefato compartilhado e, de tempos em tempos, sincronizem estas cópias. Dependendo do tempo em que um desenvolvedor fica sem sincronizar sua cópia, esta pode

[2]Concurrent Version System. Disponível em: <http://www.cvshome.org>.
[3]Rational ClearCase. Disponível em: <http://www-306.ibm.com/software/awdtools/clearcase>.
[4]Visual SourceSafe. Disponível em: <http://msdn2.microsoft.com/en-us/vstudio/aa718670.aspx>.

estar muito divergente das demais, acarretando em um esforço de convergência, culminando em retrabalho.

O problema pode surgir quando os artefatos são modificados e enviados novamente para o repositório, pois neste momento pode ocorrer um conflito. Por exemplo, dois membros de uma equipe de desenvolvimento estão trabalhando em uma mesma atividade, os dois solicitam o mesmo artefato no repositório e esse artefato está na primeira versão. Eles estão trabalhando individualmente, cada um fazendo suas alterações. Após as alterações serem efetivadas, chega o momento de submeter o artefato ao repositório. O primeiro que submeter, não terá problema. O artefato será gravado com as modificações efetuadas e então estará na segunda versão. Quando o segundo membro for submeter o artefato, ele não conseguirá, gerando assim um conflito, pois a versão que ele tinha do artefato está diferente daquela que está no repositório atualmente.

Conforme consta em [2], existem no mínimo duas formas de resolver esse problema. A coordenação das ações de cada membro do grupo pode ser realizada de forma pessimista ou otimista. Na forma pessimista, assume-se que as ações serão incoerentes e atingirão um estado inválido e indesejado. Neste caso, a ação de um membro deve bloquear as ações dos demais sobre o mesmo artefato.

Na forma otimista, assume-se que as ações serão coerentes, mesmo ocorrendo de forma isolada, e o resultado final será um estado válido. Neste caso, mais de um membro realiza ações sobre um mesmo artefato. O participante deve solicitar um bloqueio, se deseja trabalhar isoladamente sobre um determinado artefato. No entanto, o uso de bloqueios impossibilita o paralelismo de atividades que concorrem pelo artefato bloqueado. Em geral, os membros adotam a estratégia de geração de cópias do artefato, que são trabalhadas em paralelo e, posteriormente, consolidadas em uma única versão. Assim, por um lado o bloqueio evita a ocorrência de conflitos, mas reduz o paralelismo no trabalho, enquanto que o uso de cópias apenas posterga a detecção dos conflitos, mas possibilita um maior paralelismo de atividades.

Um mecanismo para reduzir o isolamento entre desenvolvedores é a percepção de mudanças entre as cópias do artefato. A informação de mudança deve ser disponibilizada para os desenvolvedores de forma a agilizar a detecção de possíveis conflitos sintáticos e semânticos, permitindo a visualização da evolução das cópias do artefato, tornando possível tomadas de decisões baseadas nas mudanças realizadas.

4 TRABALHOS RELACIONADOS

Nesta seção serão apresentadas algumas ferramentas relacionadas com o tema, encontradas na literatura, que propõem mecanismos para resolver o problema do gerenciamento de versões de artefatos.

A Ferramenta Token e LockED

Token: foi desenvolvida para apoiar o desenvolvimento concorrente de projetos de software, auxiliando na resolução dos problemas de controle de alterações nos componentes do projeto [15]. Suas principais funcionalidades são: o cadastramento dos desenvolvedores participantes do projeto, a troca de informações entre estes desenvolvedores e o controle de alterações nos componentes do projeto. Token foi desenvolvida em ambiente Linux, utilizando a linguagem de script PHP3, o banco de dados MySQL e o servidor Web Apache. A ferramenta é acessível via um navegador Internet, sendo portanto independente tanto da plataforma cliente quanto da plataforma servidora.

Essa ferramenta oferece um quadro de mensagens, indexadas por assuntos. Sempre que uma mensagem é enviada através da ferramenta Token, os desenvolvedores cadastrados recebem uma notificação por e-mail. A notificação indica o assunto da mensagem, seu tipo e contém um link para a página de mensagens do Token. Os tipos de mensagem utilizados no Token são: pergunta, resposta, informação e urgente. Os tipos de mensagem são meramente informativos, permitindo ao leitor uma

rápida identificação do objetivo da mensagem e a criação de filtros em seu e-mail. A ferramenta Token somente se preocupa com o código-fonte do sistema em desenvolvimento, não abrindo espaço para os demais produtos do processo de desenvolvimento no qual a equipe está inserida. Isto exige que o gerente da equipe procure por outras ferramentas, fazendo com que o sucesso do projeto se torne um fator de organização e comprometimento dos seus integrantes.

LockED: descreve uma abordagem que visa solucionar o problema de controle de alterações de artefatos de software no desenvolvimento distribuído de projetos de software. LockED visa controlar e impor ordem sobre a criação e as alterações de artefatos do domínio, disponibilizando-os para alocação e desalocação [24]. Seguindo o mesmo princípio da ferramenta Token, um usuário somente pode alterar um artefato quando este já estiver previamente alocado ao mesmo através da ferramenta LockED. A idéia básica é que haja uma base oficial de informações sobre o domínio em um servidor conectado à Internet/Intranet, a qual será acessada por toda a equipe. Cada integrante trabalhará em sua própria estação de trabalho utilizando a infra-estrutura de reutilização provida pelo seu ambiente de modelagem, realizando atividades de modelagem, consultas e instanciando aplicações a partir destes artefatos.

Ferramentas para Apoio à Percepção

Existem diversas propostas para apoiar a percepção em edição colaborativa. A ferramenta CO2DE [13] é um editor gráfico e síncrono de diagramas UML, baseado em uma metáfora de "máscaras", representando versões. Outro editor colaborativo de artefatos UML é D-UML [3]. Tukan [22] é um ambiente síncrono e distribuído de programação Smalltalk. O ambiente SAMS [14] permite a edição colaborativa através de interações síncrona, assíncrona e multi-síncrona. NetEdit [27] é um editor colaborativo multi-síncrono de documentos texto acessível pela Web.

A ferramenta Palantír [21] complementa sistemas de gerência de configuração, fornecendo aos desenvolvedores pertencentes à sessão de colaboração informações sobre os espaços de trabalho dos demais. Koblylinski et. al (2002) apresentam uma proposta de sistema de percepção, que permite que colaboradores monitorem atividades de outros sobre artefatos de software.

Em termos de mecanismos de percepção, CO2DE, D-UML e Tukan possuem suporte à interação síncrona, sendo que CO2DE provê também suporte assíncrono para percepção. Palantír, SAMS, NetEdit e a abordagem descrita em [10] contemplam mecanismos de percepção em interações multi-síncrona, além do suporte síncrono e assíncrono à percepção. Apoiando a percepção em modelos UML, tem-se CO2DE e D-UML.

Tukan utiliza como metáfora ícones usados em boletins meteorológicos; mudanças com alto impacto são associadas à ícones de tempo instável. Palantír fornece informações sobre severidade de mudanças sobre o artefato compartilhado através de diversas formas gráficas de representação (por exemplo, barra de progresso).

MAIS (Multi-synchronous Awareness InfraStructure)

A ferramenta está inserida no contexto do Projeto OdysseyShare [16], baseada na arquitetura cliente-servidor. Os eventos coletados são armazenados em um espaço de tuplas, tendo como implementação utilizada a especificada por JavaSpaces [23], visível a todos os desenvolvedores. Quando novos eventos são gerados, os desenvolvedores são notificados e os obtém do espaço de tuplas.

Os eventos são coletados do ambiente Odyssey, onde MAIS está inserida, e são apresentados aos desenvolvedores na forma de mensagens, descritas textualmente. Os desenvolvedores visualizam os eventos gerados, quem os gerou, além dos elementos do modelo envolvidos nos eventos. Eventos gerados pelo próprio desenvolvedor são apresentados em uma lista diferente dos gerados pelos demais.

Cada desenvolvedor pode tomar ciência sobre a ocorrência de determinado evento. Isto evita que o desenvolvedor seja alertado sobre a existência de eventos que já tenha tomado ciência em um momento anterior.

Augur: Combina Informações de Artefatos e Atividades

Augur é uma ferramenta desenvolvida por Froehlich e Dourich (2004) que permite a visualização de processos distribuídos do desenvolvimento do software. Augur gera representações visuais dos artefatos e das atividades do desenvolvimento do software e permite que os colaboradores explorem o relacionamento entre eles. O Augur é projetado para os colaboradores que participam no processo do desenvolvimento do software.

Froehlich e Dourich (2004) mostram que os artefatos podem carregar informações sobre as atividades em andamento ou concluídas, assim, tornando possível o próprio artefato extrair informações sobre as atividades que possuem dependências.

5 GERENCIADOR DE ARTEFATOS

Nesta seção serão apresentadas, o gerenciador de artefatos e como ele tratará a ocorrência de conflitos; a sua implementação; o contexto no qual serão realizados os experimentos do gerenciador de artefatos; a ferramenta Requisite e o método de avaliação do gerenciador de artefatos.

5.1 O Gerenciador de Artefatos

Os trabalhos pesquisados, na literatura, mostraram a necessidade de um gerenciador de artefatos, em um ambiente de desenvolvimento de software, devido à possibilidade de compartilhar artefatos entre os membros da equipe. Quando se pretende executar uma atividade que é realizada por um grupo de pessoas, é muito provável que surgirão alguns problemas. Esses problemas podem ser tanto de relacionamento entre os membros daequipe, como problemas na execução da mesma.

Sendo assim, durante a execução de uma atividade, podem ocorrer conflitos entre versões de um artefato. Dentre as ferramentas analisadas, na seção 4, percebeu-se que elas oferecem suporte apenas para resolver conflitos de artefatos no formato código fonte, e não oferecem um mecanismo para resolver conflitos em artefatos no formato XMI.

Além disso, na maioria das vezes, as ferramentas encontradas no mercado (NetBeans, Eclipse, SmartSVN[5], entre outras) disponibilizam mecanismos ou plugins que auxiliam o desenvolvedor à resolver os conflitos em artefatos do tipo código fonte. Os artefatos de código fonte são gerados pelo próprio desenvolvedor, o que não acontece com os artefatos no formato XMI, que são gerados por ferramentas. Por isso, os artefatos de tipo código fonte são mais fáceis de serem entendidos no momento de resolver eventuais conflitos.

Portanto resolver conflitos de artefatos no formato XMI se torna uma tarefa não trivial para os desenvolvedores.

A figura 1 simula uma situação real, onde dois membros de uma equipe estão alocados para realizar a mesma atividade. Assim, eles compartilharão os mesmos artefatos. Cada um possuirá em seu computador, no repositório local, uma cópia dos artefatos que estão no repositório central.

No repositório central estão armazenadas todas as versões dos artefatos, sendo a última versão considerada como válida. O SubVersion é uma das alternativas, encontradas no mercado, utilizada para controlar as versões dos artefatos no repositório central. Como repositório local é usado o sistema de arquivos do sistema operacional.

O gerenciador de artefatos é responsável em atender as solicitações dos usuários para acesso aos artefatos. Portanto, ele deverá fornecer facilidades, aos usuários, para manipulação dos artefatos como: requisitar um artefato, submeter um artefato, excluir um artefato, fornecer informações a

[5]The Smart Subversion-Client. Disponível em: <http://www.syntevo.com/smartsvn/>.

respeito do membro que fez a última modificação no artefato, quando foi realizada a última modificação.

De um modo geral, pode-se identificar duas situações no desenvolvimento de software.

A primeira situação (figura 1a): os desenvolvedores podem trabalhar de maneira cooperativa. Neste caso, as dificuldades para a realização de uma atividade seriam menores, pois cada um saberia o que o outro está fazendo. Entretanto, assim mesmo poderia ocorrer conflito, caso os dois realizassem modificações em uma parte comum do artefato. O ideal seria que eles cooperassem, e que só um efetivasse as modificações decididas por ambos.

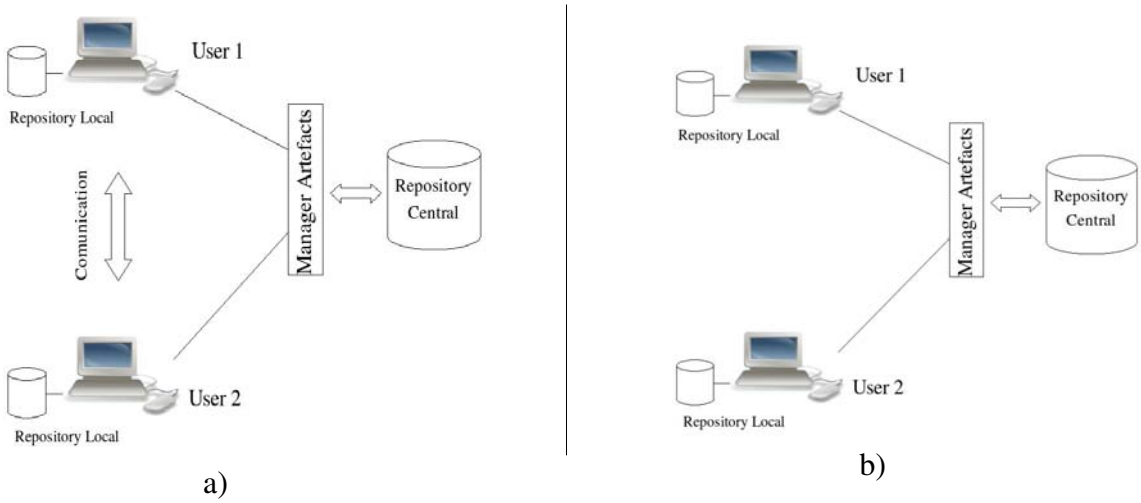


Figura 1a: Compartilhamento de Artefatos com comunicação entre os usuários. Figura 1b: Comportilhamento de Artefatos sem comunicação entre os usuários.

A segunda situação (figura 1b): os desenvolvedores podem trabalhar isoladamente, não havendo cooperação entre eles. Dessa maneira a probabilidade de haver divergência entre as versões de cada um, é maior. Neste caso, o primeiro que submeter a nova versão do artefato, ao repositório central, não terá problema. O conflito surgirá quando o segundo usuário tentar submeter a sua versão do artefato. Ele não conseguirá, pois a versão atual do artefato, no repositório central, é diferente da versão de quando ele fez a requisição do mesmo.

Uma solução apresentada, pelo gerenciador de artefatos proposto, é a utilização de bloqueio (trava) por um período de tempo para cada atividade. O tempo de bloqueio é definido pelo gerente de projeto. Essa flexibilidade é permitida em razão de cada projeto possuir suas peculiaridades. Por exemplo, uma atividade que seja mais complexa, o tempo de bloqueio pode ser aumentado para que tenha tempo suficiente para o desenvolvedor realizar as modificações necessárias.

O gerenciador de artefatos, possui também um mecanismo para resolver conflitos de artefatos no formato XMI, constituindo-se assim em uma importante contribuição. A subseção 5.4 explica como os experimentos serão realizados para validação do gerenciador de artefatos.

Uma outra funcionalidade oferecida pelo gerenciador de artefatos, ora apresentado, é permitir a combinação de informações das atividades e das versões dos artefatos. Com isso, pode-se ter conhecimento de várias informações sobre o progresso do projeto como um todo. Por exemplo, a partir do mapeamento das atividades com os artefatos, pode-se ter uma visão do progresso das atividades, quando e quais foram os desenvolvedores que tiveram participação na elaboração de um artefato em específico; para quem o artefato está alocado; quem está aguardando a liberação do artefato; quais são as atividades que dependem do artefato que está sendo produzido, dentre outros.

5.2 A Implementação

O gerenciador de artefatos está sendo implementado como um componente do ambiente DiSEN. A partir dos estudos realizados, puderam ser identificadas como funcionalidades importantes deste gerenciador:

- armazenar o artefato;
- recuperar o artefato;
- excluir o artefato;
- requisitar a trava;
- liberar a trava;
- definir tempo para alocação da trava;
- listar alterações do artefato;
- oferecer suporte para resolver conflitos.

O suporte à resolução de conflitos está sendo implementado por meio de um mecanismo e será integrado à ferramenta Requisite para os experimentos. Assim, os artefatos gerados pela ferramenta Requisite poderão ser armazenados e recuperados no repositório central. Além disso, se ocorrerem conflitos com a versão do artefato, o mecanismo deverá fornecer facilidades para que o desenvolvedor possa resolvê-los. Caso ocorram conflitos, a ferramenta apresenta, visualmente, os elementos que foram adicionados ou removidos no artefato. Os elementos que estão apresentando divergência, nas versões do artefato, serão visualizados com cores distintas, para que o desenvolvedor perceba com maior facilidade.

5.3 Contexto Considerado para Realização dos Experimentos com o Gerenciador de Artefatos

O DiSEN (Distributed Software Engineering ENvironment), um ambiente de desenvolvimento de software distribuído, incorporando a tecnologia de agentes segundo o padrão da FIPA (Foundation for Intelligent Physical Agents), utiliza a MDSODI [8], uma metodologia para desenvolvimento de software que leva em consideração algumas características identificadas em sistemas distribuídos, tais como concorrência, paralelismo, comunicação, sincronização e distribuição.

O objetivo do DiSEN é fornecer o suporte necessário para o desenvolvimento distribuído de software; a equipe poderá estar distribuída em locais geográficos distintos e trabalhar de forma cooperativa usando uma metodologia para desenvolvimento distribuído de software.

Requisite

A Requisite é uma ferramenta nativa do ADDS (Ambiente de Desenvolvimento Distribuído de Software) DiSEN que tem por objetivo auxiliar na modelagem de requisitos, prover um meio de comunicação entre os stakeholders, prover apoio à rastreabilidade e à documentação de requisitos no DiSEN [1].

As funcionalidades de criação do diagrama de caso de uso, criação de atores e casos de uso, bem como a recuperação e a persistência do modelo (salvar modelo), foram reimplementadas por Wiese (2006), o que facilita a importação e exportação de artefatos.

Na figura 2 são apresentadas as funcionalidades da ferramenta Requisite.

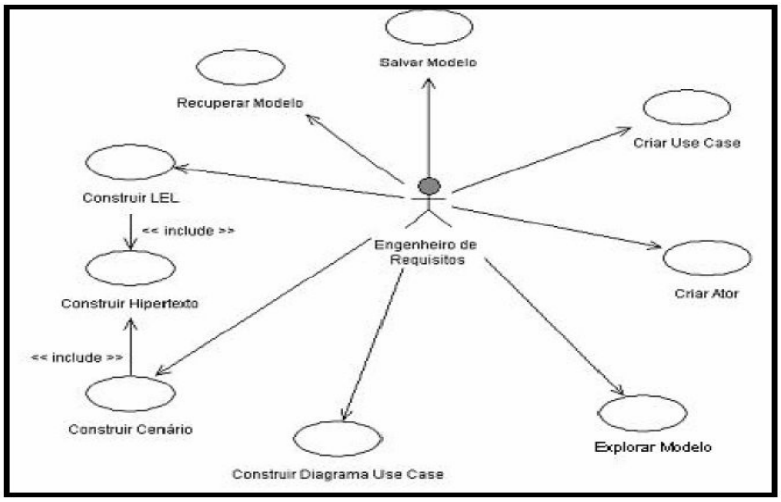


Figura 2: Funcionalidade da Requisite
Fonte: BATISTA (2003)

5.4 Método de Avaliação

A avaliação do gerenciador de artefatos será feita de acordo com a engenharia experimental [25]. O gerenciador de artefatos será integrado ao DiSEN, que oferece uma estrutura necessária para a realização dos experimentos.

A Requisite utiliza um metamodelo para a criação dos elementos no diagrama, dessa maneira, é possível mapear os objetos (elementos do diagrama de caso de uso descritos no metamodelo) e realizar neles as transformações necessárias e aplicar os filtros de importação e exportação, para a criação de arquivos XMI [26].

O gerenciador de artefatos será utilizado para auxiliar no controle das versões dos artefatos criados pelos usuários do DiSEN. Inicialmente, os experimentos estão sendo realizados utilizando o SubVersion, mas a estrutura do gerenciador de artefatos fornece uma interface para a utilização de outros repositórios de versões, por exemplo o CVS, ClearCase ou SourceSafe.

6 CONCLUSÕES

Este artigo apresenta um gerenciador de artefatos que oferece apoio à redução de conflitos. Assim, como contribuições deste trabalho pode-se destacar: facilidade para o acesso ao repositório de versões como, requisitar um artefato, submeter um artefato, mostrar a diferença entre duas versões de um artefato, exibir quando foi a última vez que um artefato foi modificado, quem fez a última modificação no artefato, quem fez mais modificações no artefato; melhor controle de alocação do artefato ao desenvolvedor (utilizando controle por tempo); a possibilidade de resolver conflitos entre versões de artefatos no formato XMI; e o uso de técnicas de *awareness,* para manter os membros da equipe atualizados sobre as modificações realizadas nos artefatos e sobre o andamento do projeto como um todo.

No contexto do desenvolvimento distribuído de software, quando um desenvolvedor requisita para trabalhar em uma atividade, o artefato é locado para o desenvolvedor daquela atividade. Quando o desenvolvedor concluir a atividade, ele desaloca o artefato. Se existisse uma atividade que fosse desenvolvida em paralelo, ou que fosse atribuída para mais de um desenvolvedor, e eles precisassem trabalhar com esse artefato, somente o primeiro que alocasse o artefato teria a permissão para alterar o artefato. Este requisito é atingido pela ferramenta LockED, ora apresentada, mas não oferece controle algum quanto a forma como ocorrem essa alocação e resolução de conflitos. O gerenciador apresentado possui este diferencial.

Para haver um melhor controle na alocação dos artefatos, o gerenciador de artefatos permite que seja controlado o tempo de alocação de um artefato para um desenvolvedor. Quando um desenvolvedor solicitar um artefato e esse já estiver alocado, o desenvolvedor irá para uma fila e assim que o artefato fosse liberado (pelo tempo excedido ou pela livre desalocação do desenvolvedor) o primeiro da fila é alocado para o uso do artefato.

Um outro problema se refere aos artefatos no formato XMI. A manipulação desse tipo de artefato pelo desenvolvedor, não é trivial, quando comparada com um artefato de código fonte. Para resolver esse tipo problema, o gerenciador de artefatos, apresenta um mecanismo para auxiliar o desenvolvedor a resolver conflitos entre duas versões de um artefato, distinguindo com cores, os objetos do diagrama de caso de uso, que estão divergentes em cada versão.

Como trabalhos futuros:

- Problemas de relacionamento entre os membros da equipe.
- Criação de um mecanismo para gerenciamento de repositórios distribuídos.
- Implementar um mecanismo para tratar conflitos de outros modelos da UML, além de use case.

Agradecimento: Os autores agradecem o Conselho Nacional de pesquisa (CNPq) pelo apoio financeiro ao projeto de pesquisa. Número do Processo: 506511/2004-9.

REFERÊNCIAS

[1]BATISTA, S. M. Uma ferramenta de apoio á fase de requisitos da MDSODI no contexto do ambiente DiSEN. 83 f. Dissertação (Mestrado) – Programa de Pós- Graduação em Informática, Universidade Federal do Paraná, Curitiba, 2003.

[2]BORGHOFF, U. M., SCHLICHTER, J. H. Computer-Supported Cooperative Work: Introduction to Distributed Applications. *Springer*, USA. 2000.

[3]BOULILA, N., DUTOIT, A. H., BRUEGGE, B. *D-Meeting: an Object-Oriented Framework for Supporting Distributed Modelling of Software*. In: Int. Workshop on Global Soft. Development, Int. Conf. on Soft. Eng., pp. 34-38, Maio, EUA. 2003.

[4]CARMEL, E. Global Software Teams – Collaborating Across Borders and Time- Zones. Prentice Hall, USA, 1999, 269p.

[5]FREITAS, A., MAIA, A., NUNES, D. Um Modelo para Interação entre Processos de Software. In: Congresso Brasileiro de Computação, CBCOMP, 4., 2004, Itajaí. *Anais*. Itajaí: Univali, 2004. p. 149 p 154.

[6]FROEHLICH, J. and DOURISH, P. Unifying Artifacts and Activities in a Visual Tool for Distributed Software Development Teams. Proceedings of the International Conference on Software Engineering ICSE 2004 (Edinburgh, UK), 387-396. 2004.

[7]HERBSLEB, J. D., MOITRA, D. Guest Editors' Introduction: Global Software Development, IEEE Software, vol. 18, no. 2, pp. 16-20, March/April, 2001.

[8]HUZITA, E. H. M., Uma Metodologia par a Desenvolvimento Baseado em Objetos Distribuídos Inteligentes. Projeto de pesquisa em andamento, Universidade Estadual de Maringá, Departamento de Informática, 1999.

[9]KIEL, L. Experiences in Distributed Development: A Case Study, In: Workshop on Global Software Development at ICSE 2003", Oregon, EUA. 2003.

[10]KOBYLINSKI, R., CREIGHTON, O., DUTOIT, A., BRUEGGE, B. Building awareness in distributed software enginering: Using issues as context. In: International Workshop on Distributed Software Development, Int. Conf. on Soft. Eng., Orlando, EUA. 2002.

[11]LANUBILE, F., DAMIAN, D., OPPENHEIMER, H. L. Global Software Development: Technical, Organizational, and Social Challenges. *ACM SIGSOFT Software Engineering Notes*. Volume 28 Number 6. November 2003.

[12]MAIR, Q. Technical Issues in the Design of a Virtual Software Corporation; ECSCW'97 OOGP workshop. 1997.

[13]MEIRE, A. P., BORGES, M. R. S., ARAÚJO, R. M. Supporting Collaborative Drawing with the Mask Versioning Mechanism. *Lecture Notes in Computer Science*, Vol. 2806, p.p. 208-223, Berlim, Alemanha. 2003.

[14]MOLLI, P., SKAF-MOLLI, H., OSTER, G., JOURDAIN, S. SAMS: Synchronous, Asynchronous, Multi-Synchronous Environments. In: 7th Int. Conf. on Computer Supported Cooperative Work in Design. CSCWD´2002, pp. 80-84, Rio de Janeiro, Brasil. 2002.

[15]MURTA, L. G. P., BARROS, M., WERNER, C. Token: Uma Ferramenta para o Controle de Alterações em Projetos de Software em Desenvolvimento, XIV Simpósio Brasileiro de Engenharia de Software, Sessão de Ferramentas, João Pessoa, outubro 2000.

[16]ODYSSEY, Projeto Odyssey. 2004. Disponível em: <http://www.cos.ufrj.br/~odyssey>. Acesso em 05 de fevereiro de 2007.

[17]OMG XMI. XMI Specification 2.1. 2005. Disponível em: <http://www.omg.org/technology/documents/formal/xmi.htm>. Acesso em 27 de Março de 2007.

[18]PASCUTTI, M. C. D. Uma proposta de arquitetura de um ambiente de desenvolvimento de software distribuído baseado em agentes. 2002, 102 f. Dissertação (Mestrado) - Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre.

[19]PRESSMAN, R. S. Software Engineering: A Practitioner's Approach. Mc Graw Hill, 5 ed., EUA, 2001.

[20]PRIKLADNICKI, R., LOPES, L., AUDY, J. L. N., EVARISTO, R.. Desenvolvimento Distribuído de Software: Um Modelo de Classificação dos Níveis de Dispersão dos Stakeholders. In: I SBSI - Simpósio Brasileiro de Sistemas de Informação, 2004, Porto Alegre. I SBSI - Simpósio Brasileiro de Sistemas de Informação, 2004. v. 1. p. 253-262.

[21]SARMA, A., NOROOZI, Z., VAN DER HOEK, A. Palantír: Raising Awareness among Configuration Management Workspaces. In: Proceedings of the 25th International Conference on Software Engineering (ICSE 2003), pp. 444-454, Maio, EUA. 2003.

[22]SCHUMMER, T.; SCHUMMER, J. Support for Distributed Teams in eXtreme Programming. In: Proceedings of eXtreme Programming and Flexible Processes Software Engineering - XP2000, Addison Wesley. 2000.

[23]SUN, JavaSpaces. Disponível em: <http://java.sun.com/developer/Books/JavaSpaces>. Acesso em 09 de março de 2007.

[24]TEIXEIRA, H. V., MURTA, L. G. P., WERNER, C. M. L. LockED: Uma Abordagem para o Controle de Alterações de Artefatos de Software, In: IV Workshop Ibero-americano de Engenharia de Requisitos e Ambientes de Software (IDEAS'01), pp.348-359, San José, Costa Rica. 2001.

[25]TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. A. G. G. Introdução à engenharia de software experimental. Relatório Técnico RT-ES-590/02. Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, 2002. 52 p.

[26]WIESE, I. S. Um Modelo de Interoperabilidade para Ambientes de Desenvolvimento Distribuído de Software. 90 f. Dissertação (Mestrado) – Programa de Pós- Graduação em Ciência da Computação, Universidade Estadual de Maringá, Maringá, 2006.

[27]ZAFFER, A., SHAFFER, C., EHRICH, R., PEREZ, M. NetEdit: A Collaborative Editor, TR-01-13, Computer Science, Virginia Tech, EUA. 2001.

Um modelo para gerenciar a comunicação em um ambiente distribuído de desenvolvimento de *software*

Daniela de Freitas Guilhermino[1 2]
danielaf@ffalm.br

Tania Fatima Calvi Tait[1]
tait@din.uem.br

Elisa Hatsue Moriya Huzita[1]
emhuzita@din.uem.br

[1]Universidade Estadual de Maringá (UEM)
Programa de Pós Graduação em Ciência da Computação
Maringá – PR, Brasil, 87020-900

[2]Faculdades Luiz Meneghel (FFALM) - Departamento de Informática
Bandeirantes –PR, Brasil, 86360-000

Resumo

Entre equipes fisicamente distribuídas prover apoio a comunicação e a tomada de decisão em grupo é fundamental para que haja cooperação e coordenação nas tarefas que necessitam ser realizadas de forma colaborativa. Abordagens técnicas e comportamentais são essenciais para a definição dos requisitos necessários ao desenvolvimento de tecnologias que auxiliem equipes geograficamente distantes. Este artigo apresenta um modelo para gerenciar a comunicação em um ambiente distribuído de desenvolvimento de *software*. O modelo propõe a comunicação síncrona, explícita e formal e define uma área de trabalho comum para gerenciar reuniões permitindo que os usuários possam interagir e encontrar um consenso para assuntos relevantes ao gerenciamento de projetos. Apresenta principalmente duas grandes funcionalidades: apoio à comunicação e edição cooperativa. Para a elaboração do modelo foram abordados aspectos relevantes relativos a Gerenciamento de Comunicação, CSCW (*Computer Supported Cooperative Work*) e Desenvolvimento Distribuído de *Software*.

Palavras-Chave: Comunicação, CSCW, *Groupware*, Desenvolvimento Distribuído de *Software*.

Abstract

Between physically distributed teams, provide support to the communication and the group decision making is basic to establish the cooperative and coordination to realize tasks that need to be carried in a collaborative way. Technical and behavior approaches are essential for the definition of the necessary requirements to the development of technologies that assist teams dispenses geographically. This article presents a proposal to manager the communication in distributed software development environment. The model proposes the synchronous, explicit and formal communication and defines an area of common work to manage meetings allowing that the users could interact and find a consensus for relevant subjects to the management of projects. It presents principally two great functions: support to the communication and cooperative edition. For the elaboration of this proposal we approached important aspects related the Management of Communication, CSCW(*Computer Supported Cooperative Work*) and Distributed Software Development.

Keywords: Communication, CSCW, Groupware, Distributed Software Development.

1 INTRODUÇÃO

A busca por redução de custos, ganho de produtividade e maior qualidade nos processos e produtos têm levado muitas organizações a distribuírem geograficamente seus recursos e investimentos. No Desenvolvimento Distribuído de *Software* (DDS) as equipes se encontram em locais físicos distintos. A separação espacial e temporal traz algumas vantagens às organizações que atuam no desenvolvimento de *software*, mas, traz também alguns desafios relacionados à comunicação, coordenação e cooperação na execução das tarefas, tais como: os níveis de distância entre os membros, diferenças culturais, fusos horários, falta de padronização de processos, incompatibilidade de ferramentas e infra-estrutura.

No trabalho cooperativo a comunicação eficaz entre os membros da equipe pode: acelerar os processos internos, facilitar a solução de problemas e de conflitos, permitir a tomada de decisão de forma mais ágil e eficiente e contribuir para a união da equipe melhorando seu desempenho. Equipes de projeto necessitam discutir suas idéias, identificar inconsistências e falhas em seus raciocínios, levantar as vantagens e desvantagens de suas estratégias, para encontrar, assim, melhores soluções para a resolução de problemas que ocorrem durante o processo de desenvolvimento de *software*. Assim, prover e facilitar a comunicação, a coordenação e a cooperação entre equipes distribuídas tem sido uma grande preocupação no gerenciamento de projetos de *software*.

O objetivo deste artigo é apresentar uma modelo que permita a comunicação síncrona, explícita e formal para um Ambiente de Desenvolvimento Distribuído de S*oftware (ADDS)*. Será apresentada uma visão geral da estrutura funcional de uma solução que permita relacionar as áreas de CSCW e DDS, visando apoiar o gerenciamento de projetos e a tomada de decisão em grupo. A tomada de decisão em grupo, neste trabalho, refere-se à participação efetiva dos membros de uma equipe no levantamento de soluções e oportunidades relevantes ao ambiente em que o grupo está inserido. O suporte a tomada de decisão em grupo, neste contexto, significa fornecer subsídios em reuniões à troca de idéias e priorização de opiniões (seleção por voto) de forma que todos participantes de vários níveis organizacionais possam contribuir igualmente.

Na seção 2 é tratado o gerenciamento de comunicação destacando sua importância no gerenciamento de projetos; na seção 3 é apresentado o DDS para contextualizar o cenário em que se insere a proposta; a seção 4 descreve CSCW e *Groupware*, assuntos relevantes ao trabalho cooperativo; na seção 5 são destacados alguns trabalhos significativos na área pesquisada; a seção 6 aborda o ambiente Disen, no qual se insere o modelo proposto para gerenciar a comunicação, na seção 7 é apresentado o modelo proposto VIMEE com suas principais características; e por fim a seção 8 apresenta as considerações finais em que são levantados os aspectos principais deste trabalho.

2 GERENCIAMENTO DE COMUNICAÇÃO

O Gerenciamento de Comunicação de projeto é definido como o conjunto de processos necessários para garantir de forma adequada a geração, a coleta, a disseminação, o armazenamento e o descarte das informações de um projeto [20]. Gerenciar a comunicação é fundamental para prover de maneira eficaz a interação entre as equipes de projeto, proporcionando a troca de informações, o compartilhamento de recursos e a coordenação dos esforços de trabalho.

O gerenciamento das comunicações envolve o planejamento, a execução e o controle do plano de comunicações. O planejamento das comunicações, segundo Martins [13], implica em determinar que informações precisarão ser geradas, para quem e como estas informações serão distribuídas. É necessário que se conheça bem todos os *stakeholders* (envolvidos ou afetados por um projeto ou por seus resultados, seja uma pessoa, grupo ou organização) e a função que cada um exerce dentro de um projeto, seu papel no processo de produção, suas responsabilidades e principalmente seu poder de decisão. Após a identificação dos *stakeholders* é necessário definir

quais são as informações relevantes para cada envolvido e, também, como e quando deverão ser comunicadas.

É de extrema importância no desenvolvimento de um projeto que as informações obtidas sejam registradas corretamente e principalmente distribuídas para os interessados. As informações de um projeto podem ser distribuídas por vários métodos, como, reuniões de projeto, distribuição de cópias de documentos, acesso compartilhado à rede eletrônica de bancos de dados, fax, *e-mail*, canal de voz e videoconferência [20].

3 DESENVOLVIMENTO DISTRIBUÍDO DE *SOFTWARE*

Com a complexidade crescente dos *softwares* desenvolvidos, o número de profissionais envolvidos no processo de desenvolvimento tem aumentando muito e a atividade exige cada vez mais criatividade, experiências e habilidades diferenciadas.

Devido à dispersão de recursos humanos capacitados, muitas organizações encontram no DDS uma alternativa para trabalhar com equipes geograficamente distantes entre si [12]. A flexibilidade e a adaptabilidade das organizações em busca de mão-de-obra especializada, a redução do prazo de entrega e o aumento de qualidade na execução de processos contribuíram para criação de ambientes de trabalho distribuído, em que os profissionais se encontram em locais diferentes realizando tarefas conjuntas.

Para Carmel [2] o DDS está causando um grande impacto na maneira como os produtos de *software* têm sido modelados, construídos, testados e entregues aos clientes. O autor destaca alguns fatores que têm contribuído para o crescimento do DDS, entre eles: custo mais baixo e disponibilidade de mão de obra; evolução e maior acessibilidade a recursos de telecomunicação; evolução das ferramentas de desenvolvimento; a necessidade de possuir recursos globais para utilizar a qualquer hora; a formação de equipes virtuais para explorar as oportunidades de mercado; a pressão para o desenvolvimento *time-to-market*, utilizando as vantagens proporcionadas pelo fuso horário diferente no desenvolvimento quase que contínuo.

Assim como existem várias motivações, existem também alguns desafios ao trabalhar com equipes distribuídas, os quais podem inviabilizar um projeto. Alguns dos principais desafios são: diferenças de língua, cultura e fusos horários; incompatibilidade de ferramentas e infra-estrutura e falta de padronização de processos. Para apoiar a colaboração em DDS são utilizadas ferramentas cuja meta é permitir o trabalho cooperativo de maneira mais produtiva, auxiliando a comunicação de idéias, compartilhamento de recursos e coordenação dos esforços de trabalho [5]. Dentre as técnicas e ferramentas de apoio, CSCW e *Groupware* são intrínsecos ao entendimento de trabalho cooperativo distribuído.

4 CSCW E *GROUPWARE*

Pela necessidade de resultados mais rápidos e de maior controle das atividades colaborativas é que surgiram as primeiras pesquisas nesta área. Em 1984 foi realizado o primeiro *Workshop* sobre o assunto, ao qual foi designado o termo CSCW [23]. CSCW, segundo Nielsen [17], pode ser definido como o estudo de técnicas, metodologias e tecnologias para apoiar o trabalho em grupo. Os recursos oferecidos por aplicações deste tipo têm o propósito de minimizar as barreiras encontradas durante o desenvolvimento de trabalhos em grupos, permitindo o aumento do rendimento na execução das tarefas em relação ao trabalho individual isolado.

O termo *Groupware* é utilizado para designar a tecnologia desenvolvida pelas pesquisas sobre CSCW [17]. *Groupware* pode ser visto como uma coleção de ferramentas computacionais, pessoas e processos de trabalho operando em sintonia em uma organização. Estas ferramentas facilitam a comunicação informal e a automatização de tarefas, permitindo a realização do trabalho em equipe de maneira mais eficaz, eficiente e criativa [7].

O *Groupware* é construído em torno de três princípios básicos: comunicação, coordenação e cooperação, por isso é referenciado como o modelo de colaboração 3C [3] [5]. De acordo com este modelo, para colaborar, as pessoas se comunicam. Durante esta comunicação, compromissos são gerados e negociados. Os compromissos são as tarefas que serão necessárias para concluir as atividades. Estas tarefas são gerenciadas pela coordenação, que organiza o grupo e garante que as tarefas sejam realizadas na ordem correta, no tempo correto e cumprindo as restrições e objetivos impostos.

As ferramentas de *Groupware*, permitem aos grupos trabalharem juntos sobre artefatos, gerenciarem seus processos, encaminharem formulários eletrônicos, compartilharem pastas e trocarem mensagens. A utilização de soluções de *Groupware* não está apoiada apenas no uso de tecnologias, mas também em mudanças de práticas organizacionais, sendo necessário, muitas vezes, a redefinição de processos de trabalho e a mudança de postura de toda equipe para criar um ambiente propício à colaboração.

Podemos encontrar cinco classes de sistemas de *Groupware* que serão descritos a seguir:

- *Sistemas de Comunicação entre Grupos:* o objetivo é, primordialmente, a comunicação entre grupos de usuários. Podem ser classificados em sistemas de comunicação síncrona e assíncrona. Sendo que a comunicação síncrona é a que necessita da coincidência de presença no mesmo espaço de tempo dos usuários; e assíncrona, que não exige a presença dos usuários ao mesmo tempo. Estas ferramentas são geralmente projetadas para serem usadas por pessoas situadas em lugares diferentes (comunicação distribuída), e em alguns casos, as ferramentas de comunicação são projetadas para complementar a interação face-a-face possibilitando o registro, e o paralelismo da comunicação [10].

- *Sistemas de Suporte a Decisão em Grupo: são* sistemas interativos baseados em computador, utilizados para facilitar a solução de problemas não-estruturados por uma equipe de trabalho cooperativo, distribuída ou não [9]. Foram desenvolvidos com intuito de melhorar a qualidade e eficácia das reuniões. Gallupe e Desanctis [6] apresentam alguns elementos facilitadores da tomada de decisão em reuniões: melhoria de pré-planejamento; melhoria da participação; atmosfera de reunião aberta e colaborativa; geração de idéias livres de críticas; organização e avaliação de idéias; objetividade de avaliação; preservação da memória organizacional e acesso à informação externa.

- *Gerenciadores de Fluxo de Trabalho (Workflows):* automatizam os procedimentos pelos quais documentos, informações e tarefas são distribuídos em algum processo de trabalho dentro da organização. Um *Workflow* pode descrever tarefas de processos de negócio em um nível conceitual necessário para compreender, avaliar e reprojetar os processos de negócios de uma organização [16]. No âmbito da Gestão de Projetos em DDS é importante integrar as técnicas de gerência com as ferramentas de *Groupware* e *Workflow* para permitir, além de uma efetiva comunicação e captura do conhecimento proveniente das interações, o monitoramento e controle das atividades envolvidas no processo de desenvolvimento.

- *Suporte Básico para Trabalho Cooperativo:* o BSCW (*Basic Support for Cooperative Work*) é um recurso de *Groupware* que fornece funcionalidades básicas para cooperação de grupos via *Internet, Intranet* ou *Extranet,* desenvolvido peto GMD-FIT (*Institute for Applied Information Technology, German National Research Center for Computer Science*) [1]. A construção deste tipo de aplicação é baseada no modelo de área de trabalho compartilhada, no qual é possível armazenar vários tipos de arquivos, e também ter acesso às ações dos membros de um grupo.

- *Sistemas de Editoração Multiusuários:* são sistemas que procuram apoiar o trabalho conjunto e simultâneo entre um grupo de pessoas em um mesmo documento. A edição de textos de forma colaborativa, pode ser feita de duas maneiras, a saber: co-autoria, onde existe mais de um autor para o documento, e cada um dos autores tem direitos similares sobre o documento; revisão, onde existe um único autor do documento e existem vários comentaristas que podem fazer

comentários sobre o texto e propor alterações, porém somente o autor pode fazer alterações efetivas no documento [21].

5 TRABALHOS RELACIONADOS

Com objetivo de contextualizar CSCW e *Groupware* no Desenvolvimento de *Software* em Ambiente Fisicamente Distribuído, o trabalho de Zanoni [24] tem uma contribuição relevante para a área de Sistemas de Informação, visto que propôs um modelo para auxiliar a colaboração entre participantes inseridos em ambientes distribuídos. Seu trabalho busca agregar abordagens importantes relacionadas a distribuição física e gerência de projetos de *software*.

Pozza [18] apresenta um modelo para cooperação que propõe flexibilizar as características que envolvem a cooperação entre usuários de *workspaces* compartilhados distribuídos, no domínio de desenvolvimento distribuído de *software*. *Workspaces* são espaços de trabalho que provêem ambientes de interação e armazenamento de informações. O modelo SPC (Sincronização Percepção Comunicação) é um modelo de cooperação para o gerenciador de *workspace* do DiSEN que aborda as características de sincronização, percepção e comunicação.

O MILOS (*Minimally Invasive Longterm Organizational Support*) fornece suporte a execução de processos e aprendizagem organizacional para o desenvolvimento distribuído de software [14]. Apoia a coordenação de projetos de engenharia de software, provê áudio e vídeo aos desenvolvedores (usando o *Microsoft NetMeeting*) e habilita o compartilhamento de linhas de códigos entre eles.

CVW (*Collaborative Virtual Workspace)* é um ambiente de computação colaborativo projetado para dar suporte a equipes de trabalho dispersas geograficamente [15]. O CVW provê um espaço virtual que consiste em aplicações, documentos, salas, recintos e prédios que permitem a interação de grupos. Do ponto de vista técnico, é um *framework* que integra diversas técnicas de colaboração (vídeo-conferência, chat, etc.).

O TeamSpace é um projeto da IBM que apóia equipes de trabalho distribuídas, modelando-as de modo mais holístico e integrado. O foco da pesquisa é prover uma reunião virtual [8], levando em consideração o fator tempo. Deste modo, habilita seus usuários para atividades passadas e presentes dos membros de seu respectivo grupo. É um sistema colaborativo baseado em comunicação via Web para gerenciar trabalhos e artefatos compartilhados. Provê comunicação síncrona e assíncrona entre diferentes membros de grupos.

6 O AMBIENTE DISEN

O DiSEN, cenário em que está inserido este trabalho, é um ADDS que busca combinar técnicas, métodos e ferramentas para apoiar todas as atividades inerentes ao processo de construção de produtos de *software*, tais como gerência, desenvolvimento e controle da qualidade [11]. Em ADDS as equipes podem estar em locais geográficos distintos trabalhando de forma cooperativa.

A arquitetura do DiSEN é constituída pelas camadas: dinâmica, de aplicação e de infra-estrutura. A camada dinâmica permite a manutenção dos componentes de *software* e serviços de forma dinâmica. A camada de aplicação dará suporte a metodologia para desenvolvimento de *software* distribuído, ao repositório para armazenamento de informações necessárias ao ambiente, aos objetos e agentes. E por fim a camada de infra-estrutura que fornece suporte às tarefas de persistência, nomeação e concorrência, e contém o canal de comunicação.

No ambiente *DiSEN* a ferramenta DIMANAGER considera funcionalidades de planejamento de projeto e controle de projeto em um ambiente distribuído [11]. A preocupação com o gerenciamento de recursos humanos também foi formalizada em um mecanismo para auxiliar o

gerente de projeto a selecionar as pessoas mais adequadas para realizar as atividades na produção de *software*.

7 UMA MODELO PARA GERENCIAR A COMUNICAÇÃO NO ADDS

Os trabalhos relacionados apresentados possuem em comum a existência de *workspaces* (áreas de trabalho) compartilhados com cooperação entre eles e o fato de não pertencerem a um domínio específico. Nas ferramentas apresentadas, exceto na CVW, a cooperação entre os desenvolvedores ocorre através de softwares já existentes, como o *Microsoft NetMeeting*, portanto a comunicação ocorre de maneira informal, sem controle de sessões e sem que haja uma coordenação explícita. Outra questão importante a ser considerada é o fato de não apresentarem recursos para agendamento das reuniões e não permitirem a elaboração de documento (ata de reunião) que registre as informações comunicadas entre os participantes.

Para a definição do modelo proposto foram consideradas algumas premissas que envolvem: trabalho cooperativo e tomada de decisão. No trabalho cooperativo a colaboração exige comunicação e coordenação. A comunicação gera compromissos e ações que devem ser executadas e conseqüentemente precisam ser coordenadas, para que se tenha maior eficiência na comunicação. Com base nos modelos de trabalho cooperativo, nas abordagens de gerenciamento de comunicação e DDS foi elaborado o modelo *Distributed Virtual Meeting* (VIMEE) que visa dar suporte a comunicação síncrona, explícita e formal, buscando apoiar o gerenciamento de projetos para equipes que se encontram fisicamente distantes.

Em uma reunião de projeto os membros podem alocar tarefas, discutir pontos importantes, compartilhar idéias e apontar sugestões sobre problemas ou oportunidades inerentes aos projetos [22]. O modelo define uma área de trabalho comum para gerenciar reuniões virtuais distribuídas em que usuários possam atuar e visualizar a atuação dos outros e apresenta principalmente duas grandes funcionalidades: apoio à comunicação e edição cooperativa. A comunicação entre os usuários ocorre de forma clara e direcionada do emissor para o receptor. Baseia-se em procedimentos formais (controle de “quem pode/deve fazer o quê e quando”) que norteiam o andamento das atividades.

A edição cooperativa compreende a elaboração de documentos, dentre os quais a ata de reunião, a exibição de documentos e imagens a todos os participantes da reunião e a determinação dos níveis de acesso às informações. Outra característica importante, é o fato de que deverá permitir, quando necessário, a visualização de artefatos de projeto por todos os participantes do evento.

Dentre as classes de Sistemas de *Groupware* o modelo proposto se caracteriza como:

- Sistemas de Comunicação entre Grupos, por permitir a comunicação em tempo real e possibilitar o registro e o paralelismo da comunicação.

- Sistemas de Suporte a Decisão em Grupo, por apresentar coordenação da comunicação de forma a promover o compartilhamento de idéias e resolução de impasses por meio de votação, favorecendo com isso, a busca por melhores soluções.

7.1 Usuários no VIMEE

A proposta faz uso dos níveis organizacionais apresentados no Modelo de Gerenciamento de Projetos de Enami [4]. Este MGP para DDS trata três níveis organizacionais (estratégico, tático e operacional) vinculando-os aos níveis gerenciais e operacionais estabelecidos para o ambiente *DiSEN*.

Os usuários do MGP nos diferentes níveis da organização são mostrados na Figura 1. No nível estratégico, o gerente geral irá executar as atividades propostas relativas ao planejamento estratégico. No nível tático estão os gerentes locais que cuidam das unidades distribuídas e os gerentes de projeto que cuidam dos projetos sob sua responsabilidade e, no nível operacional estão os engenheiros de *software* que serão responsáveis por executar as tarefas.

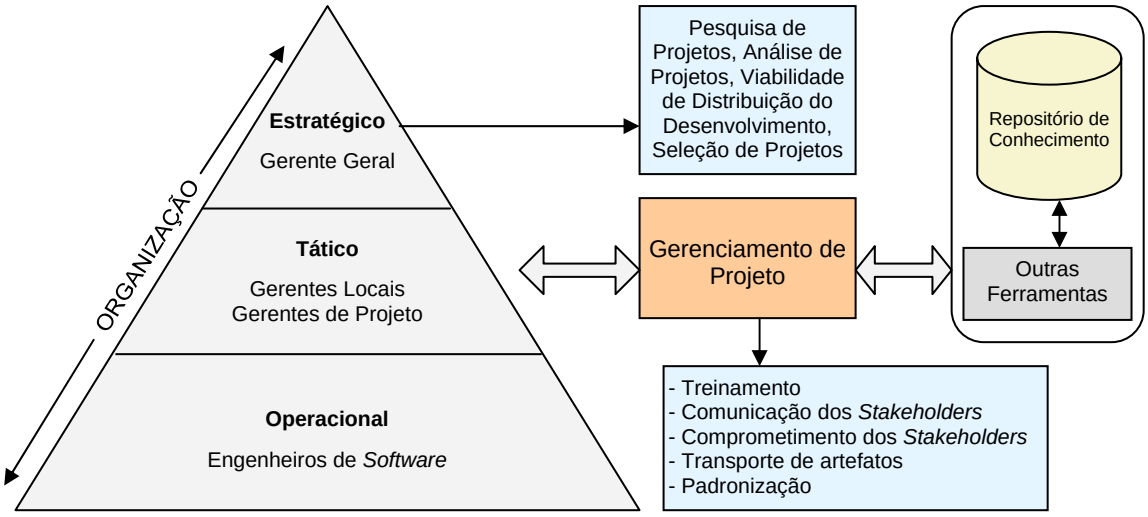


Figura 1. Componentes do Modelo de Gerenciamento.

Fonte: [4]

É importante considerar que a definição de papéis para os projetos é flexível, podendo, portanto, a cada projeto ocorrer uma troca de papéis. Por exemplo, um gerente de projeto pode ser um engenheiro de *software* em outro projeto. Os usuários definidos no MGP serão os participantes que atuarão no VIMEE, ou seja, o gerente geral, os gerentes locais, os gerentes de projeto e os engenheiros de *software*. Os papéis seguem a definição apresentada por Enami [4] para cada usuário do ambiente DiSEN:

(a) os gerentes gerais que cuidam da parte contratual com os clientes, supervisionam os gerentes de projeto e precisam de informações sobre contratos com os clientes, fornecedores, e de informações sobre o andamento dos projetos da organização para fazer a seleção dos projetos, avaliação e distribuição para as unidades geograficamente distribuídas, definindo também quais projetos devem ser priorizados, cancelados ou suspensos dentro da organização;

(b) os gerentes locais que são os gerentes de cada unidade distribuída e que precisam de informações para gerenciar os RH e materiais disponíveis para a sua unidade determinando quais recursos da sua unidade estão disponíveis para cada projeto, supervisionando os projetos alocados em sua unidade e se preocupando em motivar as pessoas pois, são os que mantêm maior relacionamento face a face com os participantes do local;

(c) os gerentes de projeto que necessitam de informações para o planejamento e controle dos projetos sob sua responsabilidade;

(d) os engenheiros de *software* que precisam de informações sobre sua agenda para executar as atividades de um projeto.

7.2 Especificação Funcional do VIMEE

A especificação funcional se desenvolveu a partir das características do modelo de colaboração 3C e do modelo SPC [18], que envolvem comunicação, coordenação e cooperação. Dentro destas características foram levantados aspectos relacionados à metodologia para reuniões, tais como os usuários fundamentais ao processo, seus papéis e funções conforme apresentado no Quadro 1.

Em relação a coordenação das atividades, atribuiu-se o papel de mediador a um dos participantes da reunião, que controlará o tempo da reunião, o tempo de uso da palavra de cada participante. Outras funções do mediador são apresentadas no Quadro 1. É importante ressaltar que o tempo da reunião e de cada participação será préestabelecido no agendamento da reunião. Com relação à edição cooperativa, deverá dar suporte ao trabalho conjunto e simultâneo entre os participantes em um mesmo documento, no caso, a ata de reunião. A primeira versão da ata será elaborada pelo Secretário (eleito pelos participantes no início da reunião) e deverá ser apreciada pelos participantes que poderão contribuir com sugestões e correções. A versão da ata definitiva será produzida quando todos participantes considerarem o documento adequado.

Papel	Responsáveis	Funções
Solicitante	Podem solicitar reuniões: Gerente Geral, Gerentes Locais e Gerentes de Projeto	- Agendar / Desmarcar reuniões
		- Definir participantes
		- Definir pauta
		- Definir mediador
Secretário	Qualquer um dos usuários convocados para reunião. Será eleito pelos participantes no início da reunião	- Elaborar a ata
		- Submeter ata para apreciação
Mediador	Solicitante ou um dos usuários escolhidos pelo solicitante	- Iniciar reunião
		- Controlar gravação
		- Iniciar e encerrar as sessões
		- Controlar lista (uso da palavra)
		- Chamar atenção para item de pauta
		- Encerrar reunião
Participante	Qualquer um dos usuários convocados para reunião: Gerente Geral, Gerentes Locais, Gerentes de Projeto e Engenheiros de *Software*	- Visualizar participantes
		- Eleger secretário
		- Pedir palavra
		- Fazer uso da palavra
		- Votar em caso de impasses
		- Consultar artefatos de projeto
		- Exibir documentos e/ou imagens
		- Apreciar documentos e/ou imagens
		- Apreciar ata

Quadro 1. Papéis no modelo VIMEE.

7.3 Especificação dos Fluxos das Tarefas

Esta seção apresenta os fluxos das tarefas, que demonstram a seqüência de passos necessários para que se possa de acordo com o conjunto de regras definidas atingir a automação dos processos que fazem parte do modelo VIMEE que será apoiado por uma ferramenta.

O VIMEE possui 3 seções: *Agendar Reunião*, *Consultar Reunião* e *Realizar Reunião Virtual*, conforme mostra a Figura 2. A seção *Agendar Reunião* permite ao solicitante agendar uma reunião e definir os elementos necessários para a execução da reunião (participantes, mediador, pauta). A seção *Consultar Reunião*, permite aos membros de um projeto consultarem as informações à respeito das reuniões agendadas. E por fim, a seção *Realizar Reunião Virtual* fornece acesso a área colaborativa de execução das reuniões.

Os usuários de todos os locais (unidades distribuídas) podem ter acesso ao VIMEE. Vale salientar que cada projeto pode ter participantes em vários locais. A Figura 2 apresenta um exemplo de dois projetos (A e B) distribuídos em unidades locais distintas. Para o projeto A foram alocados

os usuários A1, A2 e A3, sendo que o gerente do projeto A é o usuário A1. Para o projeto B foram alocados os usuários B1, B2 e B3, sendo que o gerente do projeto B é o usuário B1. O gerente geral é o gerente do local 2.

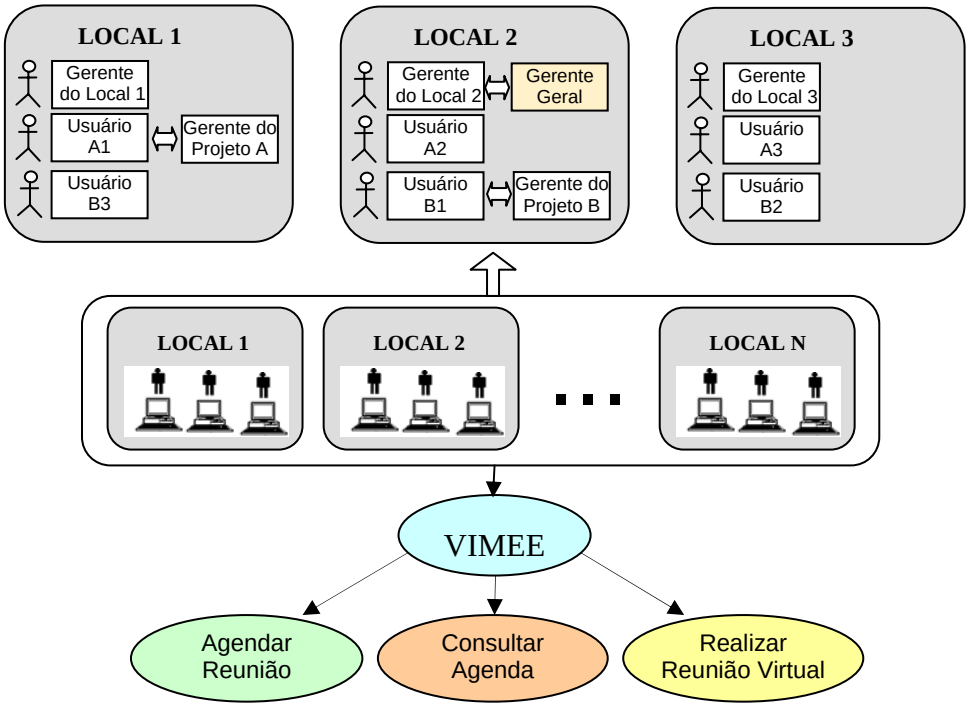


Figura 2. Seções do VIMEE.

A Figura 3, referente à seção *Agendar Reunião* descreve o fluxo das principais tarefas para agendamento de uma reunião, desde a solicitação por um dos usuários até a concretização do agendamento. Ao concluir o processo, as informações de agendamento deverão ser persistidas para posteriores consultas e os convocados deverão ser comunicados do compromisso estabelecido.

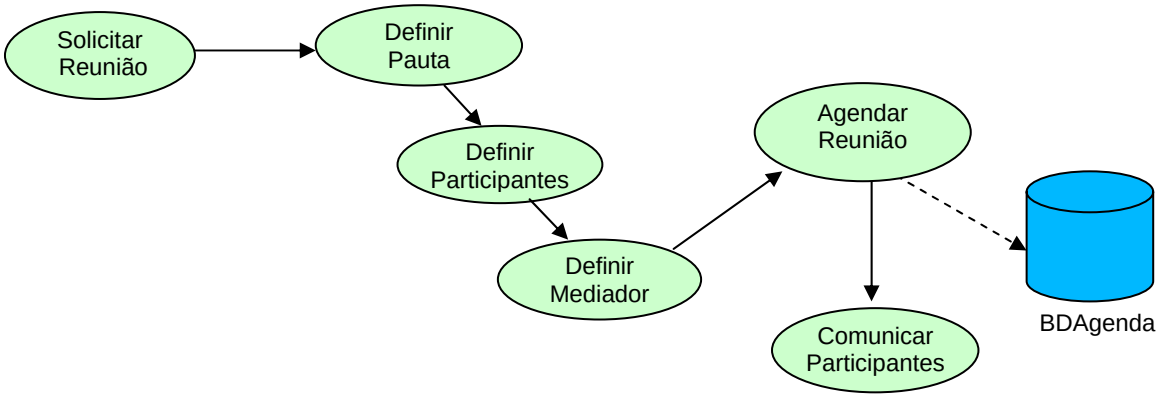


Figura 3. Fluxo de tarefas da seção Agendar Reunião.

A seção 3, que apresenta a Reunião propriamente dita, pode ser analisada considerando 3 visões: do participante, do secretário e do mediador. Ao Confirmar a presença na reunião o usuário, anteriormente convocado, deverá eleger o secretário. A Figura 4 apresenta o fluxo de tarefas da seção *Reunião Virtual* na visão do participante.

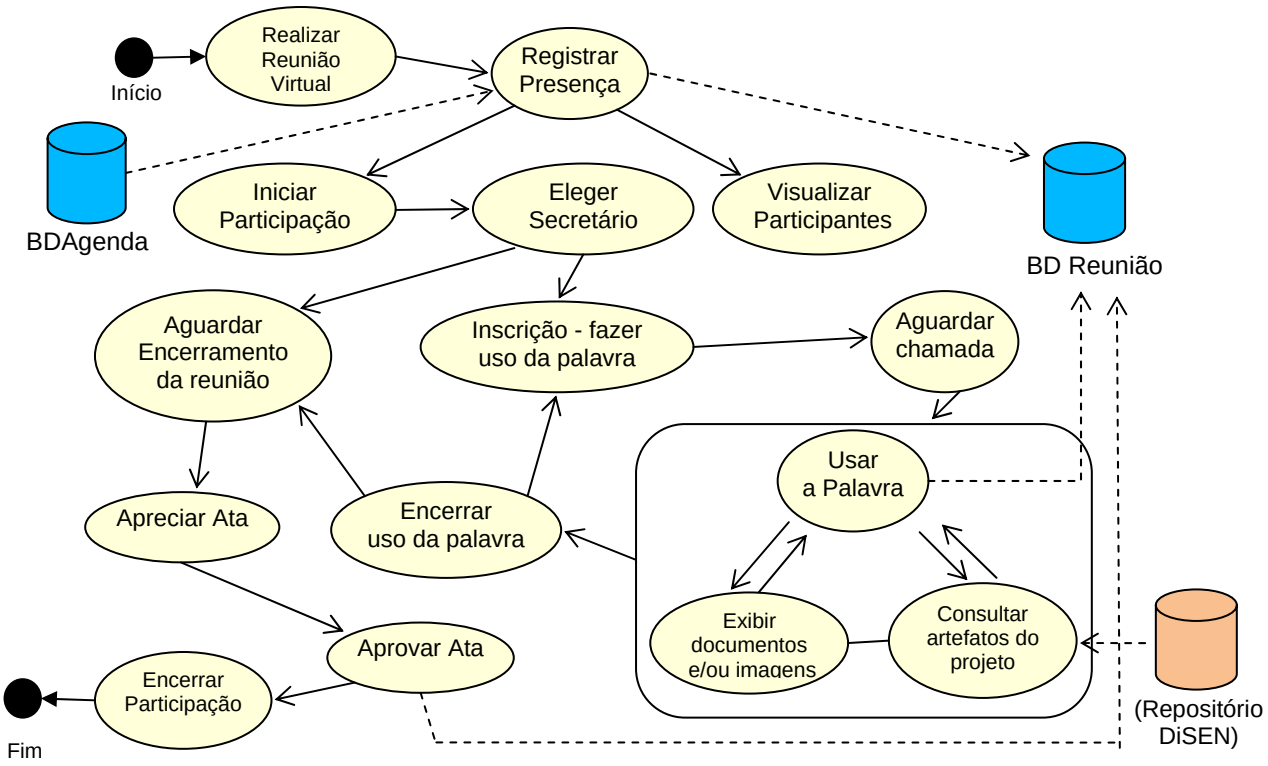


Figura 4. Fluxo de tarefas da seção Realizar Reunião Virtual – Visão do participante.

O participante tem a percepção de todos os membros que participam da reunião, ao iniciar sua participação deve votar para que seja definido o secretário, que será o responsável pela ata da reunião. Para fazer uso da palavra o participante deve efetuar sua inscrição e aguardar o momento de se expressar. O participante pode exibir documentos e/ou imagens e consultar artefatos de um projeto armazenado no repositório do ambiente, o que implica que a ferramenta apresenta integração com o ambiente permitindo o acesso as informações do DiSEN. Alguns assuntos discutidos na reunião poderão necessitar de votação para que haja um consenso na escolha de uma melhor solução, portanto, a ferramenta deve prover mecanismos para permitir a seleção de uma opção por meio de voto. Após a discussão de todos os itens de pauta, o participante deverá apreciar a ata de reunião, elaborada pelo secretário, e poderá dar a sua contribuição para que então possa ser aprovada a ata e por fim encerrada a reunião.

7.4 Desafios à comunicação no ADDS

A comunicação para ADDS apresenta alguns desafios significativos, tais como: a usabilidade, a documentação das reuniões e a segurança de dados. É preciso considerar os desafios que a dispersão geográfica impõe à comunicação para que ela possa efetivamente auxiliar o gerenciamento de projetos, contribuindo, dessa forma, para a obtenção de qualidade nos processos e produtos de *software*.

Quando usuários de diferentes culturas interagem, surgem dificuldades relacionadas principalmente a idiomas e fusos horários, os quais podem provocar problemas de usabilidade e acessibilidade. O fuso horário é um grande desafio também quando se trata de comunicação síncrona, pela necessidade da coincidência de presença no mesmo espaço de tempo. Neste aspecto, estabelecer horários que sejam mais adequados aos encontros virtuais, considerando todas as unidades locais, torna-se essencial. É preciso encontrar um ponto de equilíbrio entre os objetivos de usabilidade e acessibilidade com o perfil do participante para que o sistema cumpra seu papel da

forma mais agradável possível e para que seja garantida a valorização do usuário em seu trabalho [19].

É preciso reconhecer que a comunicação gera novos compromissos e ações a serem executadas as quais poderão interferir na continuidade dos projetos, como, por exemplo, a redefinição de procedimentos, a realocação da equipe e de recursos materiais e a resolução de conflitos. Neste contexto o VIMEE permite armazenar o documento gerado pela reunião (ata) de forma que ele possa realimentar com ações corretivas o ambiente e seus projetos. Todas essas ações tratadas no processo de comunicação devem ser disseminadas no ADDS. Outro aspecto importante considerado é a segurança nos dados, para o qual o modelo define uma política de segurança determinando níveis de acesso às informações, consideração essencial para eficiência e eficácia do processo.

8 CONSIDERAÇÕES FINAIS

O DDS implica na interação de muitas atividades e criação de muitos artefatos, o que traz maior complexidade para a coordenação dos processos e equipes. O ADDS provê suporte ao desenvolvimento e ao gerenciamento de processos de *software*, considera aspectos técnicos e comportamentais que são essenciais para a cooperação entre equipes geograficamente distribuídas.

Este artigo apresentou um modelo para apoiar a comunicação síncrona, formal e explícita no ADDS, buscando contribuir ao gerenciamento de projetos, gerenciamento de comunicações e a tomada de decisão em grupo. Aspectos relacionados ao gerenciamento de comunicação, DDS e CSCW foram fundamentais na elaboração do modelo VIMEE.

De acordo com o que foi apresentado na seção 7 o modelo proposto cobre alguns aspectos não observados em soluções existentes, a saber: a comunicação entre os usuários ocorre de forma clara e direcionada do emissor para o receptor; baseia-se em procedimentos formais que norteiam o andamento das atividades, apresentando o controle de "quem pode/deve fazer o quê e quando"; permite a elaboração de documentos de forma cooperativa, dentre os quais a ata de reunião; permite a exibição de documentos e imagens a todos os participantes da reunião e provê a determinação dos níveis de acesso às informações. Outra característica importante, é o fato de permitir a visualização de artefatos de projetos por todos os participantes do evento, apresentando portanto integração com ADDS. Dentro destas características foram levantados aspectos relacionados a metodologia para reuniões, levando em consideração a atribuição dos papéis e a função de cada membro no processo.

Agradecimento

Ao CNPQ pelo apoio financeiro. Processo nr. 50651111/2004-9

REFERÊNCIAS

[1] BROOKE, J. User interfaces for CSCW systems, in CSCW in practice : an Introduction. and case studies, Dan Dapier e Colston Sanger (eds.) Springer-Verlag:1993.

[2] CARMEL, E. Global Software Teams – Collaborating Across Borders and Time - Zones. Prentice Hall, USA, 1999, 269p.

[3] ELLIS, C.A., GIBBS, S.J., and REIN, G.L. Groupware – Some Issues and Experiences. Comm. of ACM, 34(1), 1991, 38-58.

[4] ENAMI, L.N.M., Um Modelo de Gerenciamento de Projetos Para um Ambiente de Desenvolvimento Distribuído de Software. Dissertação (Mestrado em Ciência da Computação) - Departamento de Informática. Maringá-Pr: Universidade Estadual de Maringá, 2006.

[5] FUKS, H., GEROSA, M. A., and LUCENA, C. J. P, The Development and Application of Distance

Learning on the Internet. The Journal of Open and Distance Learning, 17(1), 2002, 23-38.

[6] GALLUPE, B.R. e DESANCTIS, G. Computer-Based Support for Group Problem-Finding: an Experimental Investigation, MIS Quarterly, 12, 2 (1988), 277-296.

[7] GEROSA, M.A., RAPOSO, A.B., FUKS, H., LUCENA, C.J.P, "Combinando Comunicação e Coordenação em Groupware", 3ª Jornada Ibero-Americana de Engenharia de Software e de Conhecimento – JIISIC 2003, Anais Eletrônicos, 26-28 de Novembro, Valdivia, Chile.

[8] GEYER, W., RICHTER, H., FUCHS, L., FRAUENHOFER, T., DAIJAVAD, S., POLTROCK, S., A Team Collaboration Space Supporting Capture and Access of Virtual Meetings, **ACM 2001 International Conference on Supporting Group Work**, Boulder, Colorado, USA, 2001.

[9] GROBOWSKI, R., MCGOFF, C., VOGEL, D., MARTZ, B. e NUNAMAKER, J. Implementing Electronic Meeting Systems at IBM', MIS Quarterly, pp.369–384. (December 1990).

[10] GUNNLAUGSDOTTIR, J., Seek and you will find, share and you will benefit: organising knowledge using Groupware systems. International journal of Information Management 23 (2003) 363-380.

[11] HUZITA, E.H.M., TAIT, Tania F.C. ; PEDRAS, M. E. V. & SANTIAGO, G. P. (2004) DIMANAGER: A Tool for Distributed Software Development Management. In: International Conference on Enterprise Information Systems, Portugal, p.659-662.

[12] KIEL, L., Experiences in Distributed Development: A Case Study, In. Workshop on Global Software Development at ICSE, Oregon, EUA. Proceedings, 2003.

[13] MARTINS, J.C.C., Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML. 1. ed. Rio de Janeiro: BRASPORT, 2004.

[14] MAURER, F., MARTEL, S., Process Support for Distributed Extreme Programming Teams. Proceeding of the 24th Internacional Conference on Software Engineering, Internacional Workshop on Global Software Development, Orlando, 2002.

[15] MAYBURY, M., Collaborative Virtual Environments for Analysis and Decision Support. Communications of the ACM, 2001, p. 51-54.

[16] MOECKEL, A., Desmistificando o BSCW. Curitiba: CEFET-PR, 2001.21p. Apud Bentley, R. Supporting collaborative information sharing with the word widemweb: the BSCW shared workspace system. Boston, 1995.

[17] NIELSEN, J.; Multimedia and Hypermedia – The Internet and Beyond, Academic Press Inc., 1996.

[18] POZZA, R., Proposta de um modelo para cooperação baseado no gerenciamento de workspace no ambiente DiSEN. Dissertação (Mestrado em Ciência da Computação) - Departamento de Informática. Maringá-Pr: Universidade Estadual de Maringá, 2006.

[19] PREECE, J. Design de Interação: além da interação homem-computador. Porto Alegre: Bookman, 2005.

[20] PROJECT MANAGEMENT INSTITUTE (PMI). A Guide to the Project Management Body of Knowledge- PMBOK, 3a edição, 2004.

[21] REIS, R.Q., Uma Proposta de Suporte ao Desenvolvimento Cooperativo de Software no Ambiente PROSOFT. Tese de Mestrado. Porto Alegre: PPGC-UFRGS, 1998.

[22] SILVEIRA, M.C.; TAIT, T.F.C.; CYBIS, W., Ergonomic Process Development of Computer Supported Cooperative Work. In: Global Ergonomic Conference, 1998, Cape Town - Africa do Sul. Proceedings of the Ergonomic Conference. EUA : Elsevier Science LTD, 1998.

[23] WILSON, P. Computer supported cooperative work: an introduction. Oxford: Intellect, 1991.

[24] ZANONI, R., CSCW e Groupware: contexto no desenvolvimento de software em ambiente fisicamente distribuído. Simpósio Brasileiro de Sistemas de Informação (SBSI), Florianópolis, 2006.

Un Entorno de Decisión para la Gestión y Evaluación de la Calidad en el Desarrollo del Software

Peláez J.I. Doña J.M
Dpto. de Lenguajes y Ciencias de la Computación
Universidad de Málaga. Málaga 29071. España
{jignacio, jmdona}@lcc.uma.es

Dapozo G. N. Greiner C. L.
Dpto. de Informática. Universidad Nacional del Nordeste.
9 de Julio 1449 (3400) Corrientes, Argentina
{gndapozo, cgreiner}@exa.unne.edu.ar

Abstract

Actually, companies use software environments to manage critical processes. For this reason, the control of quality in software products has a main role in the manufacturing of the software programs. The methodologies of software development and the models of capacity contribute to improve the quality and provide global measures of effectiveness in the practice of Software Engineering. These models are oriented to the evaluation of the development processes and maintenance of systems and software products. In this paper, a decision environment for evaluation of software projects is presented. This tool allows evaluating software products to find the weaknesses and strengths of running projects. Also this tool is able to calculate the effectiveness of the different tasks of the software methodology development.

Keywords: Software quality, evaluation methodologies, quality evaluation tools.

Resumen

Actualmente se ha extendido el uso, por parte de las empresas, de entornos software para controlar sus procesos más críticos. Por esta razón, el control de la calidad en los procesos de producción de productos software ha pasado a ser vital. En esta área, las metodologías para el desarrollo de software y los modelos de calidad contribuyen a mejorar la calidad aportando medidas globales de efectividad en la práctica de la Ingeniería del Software. Estos modelos están orientados a la evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos software. En este trabajo se presenta un entorno de decisión para la evaluación de proyectos software. Esta herramienta permite evaluar los productos software para encontrar las debilidades y fortalezas de los proyectos en curso. Además permite estimar la efectividad de las diferentes fases y tareas de las metodologías de desarrollo del software.

Palabras claves: Calidad de software, metodologías de evaluación, herramientas de evaluación de calidad.

1 Introducción

La calidad en los productos software es una exigencia creciente, dado que cada vez es más amplio el uso del software en procesos que son críticos para las organizaciones. Por otra parte, el software se ha convertido en un elemento crucial en la economía mundial.

Una definición amplia de calidad, planteada en la norma UNE-EN ISO 8402 (1995), expresa que "la calidad es el conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas". Llevada esta definición al campo de la ingeniería de software, la IEEE (Std 610, 1990) señala que "la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario".

Por su parte Pressman (2004) se refiere a la calidad del software como "la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente".

Las definiciones anteriores resaltan tres puntos importantes:

- Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad.
- Los estándares especificados definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se siguen esos criterios, casi siempre habrá falta de calidad.
- Existe un conjunto de requisitos implícitos que a menudo no se mencionan. Si el software se ajusta a sus requisitos explícitos pero falla en alcanzar los requisitos implícitos, la calidad del software queda en entredicho.

Se debe tener en cuenta a la hora de abordar la calidad en el software que el desarrollo del mismo es de naturaleza compleja, dado que tiene un conjunto de características que lo hace un producto peculiar:

- Se desarrolla, no se fabrica en el sentido clásico de la palabra.
- Se trata de un producto lógico, intangible, sin existencia física.
- No se degrada con el uso.
- Por la complejidad del software y la ausencia de controles adecuados, se suele entregar conscientemente con defectos (incluso públicamente declarados).
- Un gran porcentaje de la producción se hace aún a medida en vez de emplear componentes existentes y ensamblar.

- Es muy flexible. Se puede cambiar con facilidad e incluso reutilizar fragmentos.

Sommerville (2005), afirma que la mejor forma de asegurar que los requerimientos de calidad sean verificables, es expresándolos cuantitativamente. Esto puede lograrse midiendo algunas características del software que están relacionadas, de alguna manera, con el sistema deseado.

Sin embargo, es necesario señalar que el proceso de aseguramiento de la calidad del software (SQA-Software Quality Assurance) es una actividad que se aplica a lo largo de todo el proceso de ingeniería de software. Si se realiza un proceso de desarrollo con calidad, es de esperar que se obtenga un producto de calidad. El aseguramiento de la calidad se realiza a través de un proceso de desarrollo guiado por modelos y metodologías que permiten llevar adelante procesos con final predecible.

Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Las metodologías de desarrollo de software contribuyen a mejorar la calidad.

Los programadores tradicionales argumentan que la aplicación de una metodología supone una gran carga. Es cierto, pero si no se emplea una metodología pueden surgir los siguientes problemas:

- Resultados impredecibles
- Detección tardía de errores
- Cambios de organización que también afectarán al proceso
- Resultados distintos con nuevas clases de productos

El desarrollo del software se suele organizar en proyectos. En cada proyecto de desarrollo se deben aplicar las directrices de calidad fijadas a nivel de la organización. Para ello es imprescindible la adaptación de las mismas a las condiciones de cada proyecto. Las directrices contenidas en el sistema de calidad deben adecuarse a cada uno de los proyectos.

En este trabajo se presenta una herramienta de software desarrollada en la Universidad de Málaga para evaluar la calidad a nivel de los proyectos en curso, de los proyectos terminados y a nivel de la empresa. La primera opción, proporcionará los métodos para evaluar la calidad de los proyectos en curso, las evaluaciones obtenidas permitirán determinar el grado evolución de la calidad y finalmente, la calidad a nivel de empresa, permitirá realizar una evaluación bajo el marco de trabajo CMMI-SW.

El artículo se estructura como sigue: en la siguiente sección se introduce la metodología DUM y el modelo de madurez para el desarrollo de sistemas de calidad; a continuación se presenta una nueva herramienta para la medición de la calidad y finalmente se exponen las conclusiones y futuras líneas de trabajo.

2 Metodología DUM

DUM (Diseño Unificado con Métrica) es una metodología evolutiva e incremental de desarrollo del software que ha sido creada en el departamento de Lenguajes y Ciencias de la Computación de Universidad de Málaga (Peláez et al. 2007). Basada en un enfoque iterativo incremental, esta metodología realiza una especificación exhaustiva de todas las actividades y tareas que se realizan en las diferentes fases, prestando especial atención por alcanzar un nivel superior de madurez según el marco CMMI/Carnegie Mellon.

Sus características principales son:

- Proporcionar una guía para ordenar las actividades de un equipo.
- Dirigir las tareas de cada desarrollador por separado y del equipo en conjunto.
- Especificar los artefactos que deben desarrollarse.
- Ofrecer criterios para el control y la medición de los productos y actividades del proyecto.

La metodología DUM esta dividida en 6 fases: Preliminar, Inicio, Elaboración, Construcción, Transición y Mantenimiento. Cada una de las fases se ejecuta mediante una o más iteraciones. Cada iteración esta compuesta por actividades pertenecientes a alguna de las siguientes líneas de trabajo:

- Planificación
- Estudio de viabilidad
- Identificación y especificación de requisitos
- Análisis del Sistema
- Diseño del Sistema
- Construcción del Sistema
- Prueba del Sistema
- Mantenimiento del Sistema
- Labores de Calidad
- Labores de Seguridad
- Labores de Gestión de la Configuración
- Labores de Gestión de Proyectos

La descripción detallada de estas iteraciones y de las actividades que las componen se puede encontrar en Peláez et al. (2007). Las actividades de dichas líneas de trabajo a su vez estarán compuestas por tareas. Cada una de estas tareas deberá ser ejecutada en función del correspondiente orden y fechas dispuestas en la planificación realizada. Además, en cada tarea se generará un documento y una serie productos directos de su ejecución.

Modelo de Madurez

El objetivo primordial de la ingeniería de software es producir un sistema, aplicación o producto de alta calidad. Para lograr este objetivo, los ingenieros del software deben aplicar métodos efectivos junto con herramientas modernas dentro del contexto de un proceso maduro de desarrollo de software. Además un buen ingeniero de software debe medir si la alta calidad se lleva a cabo (Pressman, 2004).

El CMMI (*Capability Maturity Model* - Modelo de Madurez de Capacidad), proporciona una medida de efectividad global de las prácticas de Ingeniería del Software. Es un modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software.

Este modelo nace como resultado del esfuerzo del Instituto de Ingeniería del Software (SEI) de la Universidad Carnegie Mellon, que culmina en el año 2002. El objetivo es unificar los modelos de ingeniería de software (SW-CMM, también conocido como CMM), de ingeniería de sistemas (SECMM) y de desarrollo integrado de productos (IPD-CMM).

Este modelo establece un conjunto de prácticas o procesos clave agrupados en Áreas Clave de Proceso (KPA - *Key Process Area*). Para cada área de proceso define un conjunto de buenas prácticas a través de las cuales una organización progresivamente alcanza mayor calidad, productividad y menores costos en el desarrollo de software.

CMMI tiene como objetivo el mejoramiento continuo de la calidad de los procesos y productos de una organización. A través de la evaluación de los procesos de una organización, es posible determinar una magnitud denominada Nivel de Madurez del Proceso.

3 Descripción de la herramienta

En esta sección se presenta una herramienta de software diseñada para evaluar la calidad a nivel de los proyectos en curso, de los proyectos terminados y a nivel de la empresa, donde se combinan tanto el modelo de madurez como la metodología de diseño unificado con métrica DUM.

Este sistema ha sido desarrollado con el lenguaje de programación Delphi utilizando una base de datos SQL Server. Además, forma parte de un proyecto mayor que integra funciones de planificación, documentación y seguridad que permitirá dar soporte a todas las actividades críticas de la gestión de proyectos de software.

Aplicación para la evaluación de la calidad

Al iniciar la aplicación, el interfaz muestra tres opciones que hacen referencia a los tres niveles citados anteriormente: Calidad a Nivel de Proyecto, Evolución de la Calidad y Calidad a Nivel de Empresa.

Figura 1: Selección de opciones

Calidad a nivel de empresa

Si seleccionamos la primera opción, accedemos a la interfaz de Calidad a nivel de empresa. Esta interfaz nos permite generar una nueva evaluación teniendo acceso a los datos básicos de identificación de la evaluación como del modelo y alcance deseado.

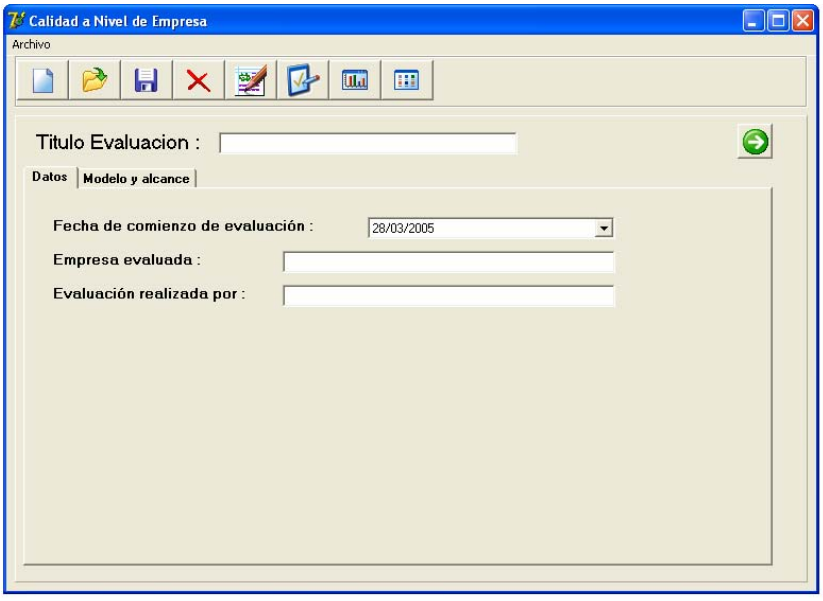


Figura 2: Nueva evaluación. Datos de Identificación

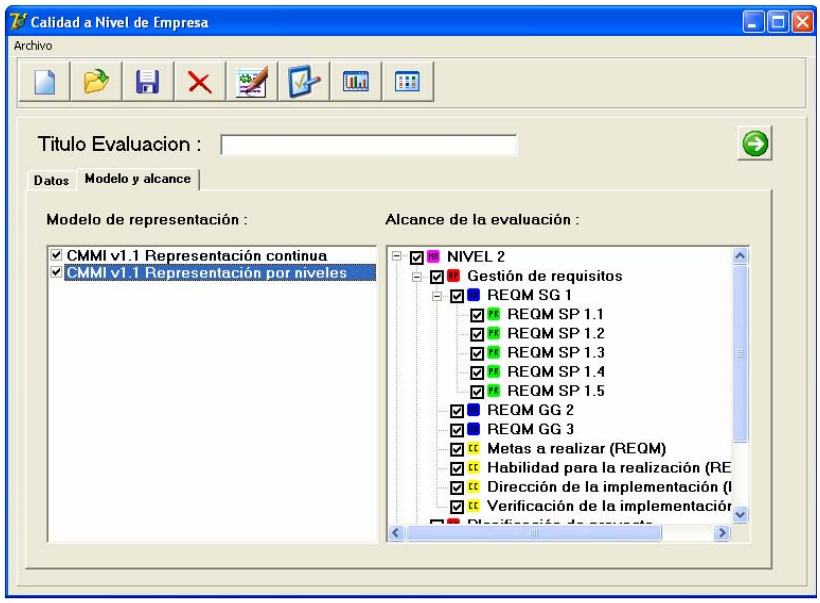


Figura 3: Selección del modelo y alcance de la evaluación. Representación por niveles

En la pestaña Modelo y alcance, se puede seleccionar el tipo de representación elegida para el modelo, optando entre representación continua, por niveles o ambas. De igual manera se podrá seleccionar el alcance de la evaluación, seleccionando fácilmente qué prácticas, objetivos, áreas de proceso y/o niveles se desea evaluar para el modelo por niveles, siendo idéntica la selección del alcance para el modelo continuo.

Una vez seleccionado el tipo y alcance de la evaluación se puede pasar a la fase de recopilación de artefactos, ya sea pulsando sobre el botón Siguiente situado en la parte superior derecha, o sobre el quinto botón de la barra de botones.

Se visualiza el siguiente formulario:

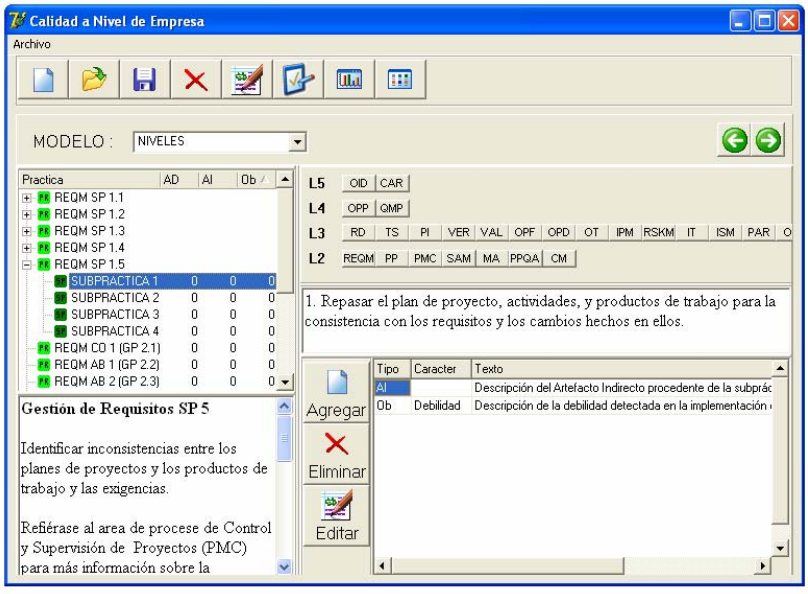


Figura 4: Recopilación de artefactos

Fácilmente se puede ir cambiando entre los dos modelos de representación elegidos. En la parte superior derecha tendremos un listado de botones con las distintas áreas de procesos de la que consta cada representación. Una vez pulsado uno de estos botones, en el lado izquierdo se desplegará un árbol compuesto de las distintas prácticas y subprácticas de la que consta cada área de proceso. Seleccionando uno de sus nodos, en la parte inferior al árbol aparecerá una descripción de la práctica, complementándose con una descripción de la subpráctica si de este tipo fuera el nodo seleccionado.

En la parte inferior derecha se encuentra situado un listado de los distintos objetos que comprende cada práctica o subpráctica, los que se pueden eliminar, editar o agregar elementos nuevos, si este fuera el caso. Al introducir un nuevo artefacto, se puede elegir el tipo y el carácter, introduciendo además una descripción sobre el mismo.

Terminada la introducción de datos, se puede proceder a la evaluación de los mismos. Para ello, se debe pulsar el botón de Evaluación  de la interfaz principal, desplegándose la siguiente ventana:

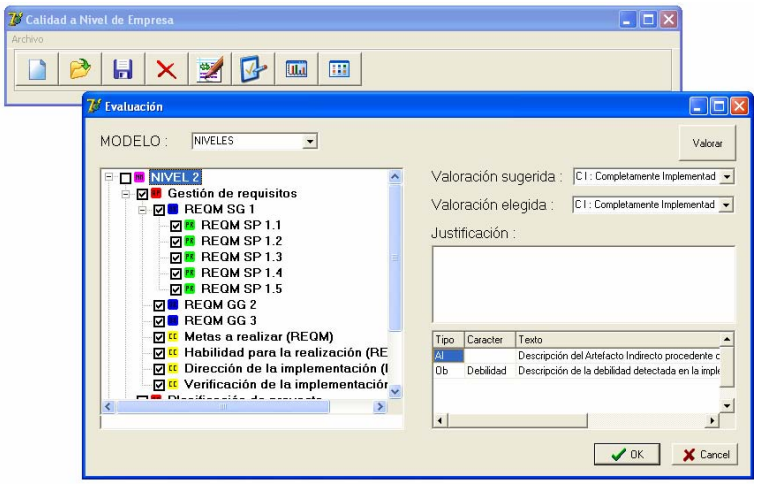


Figura 5: Evaluación del proyecto

En su lado izquierdo, y dentro del alcance seleccionado, se selecciona el aspecto a evaluar. Una vez realizada la selección, sólo tendremos que pulsar sobre el botón Valorar para obtener la valoración sugerida. Si bajo el criterio del usuario, la valoración elegida no fuera la misma que la valoración sugerida, deberá dar una justificación en el campo dedicado para ello.

Una vez realizada la valoración de los datos, podremos acceder mediante el botón  de la interfaz principal a la representación gráfica de los resultados obtenidos como se muestra en los siguientes ejemplos:

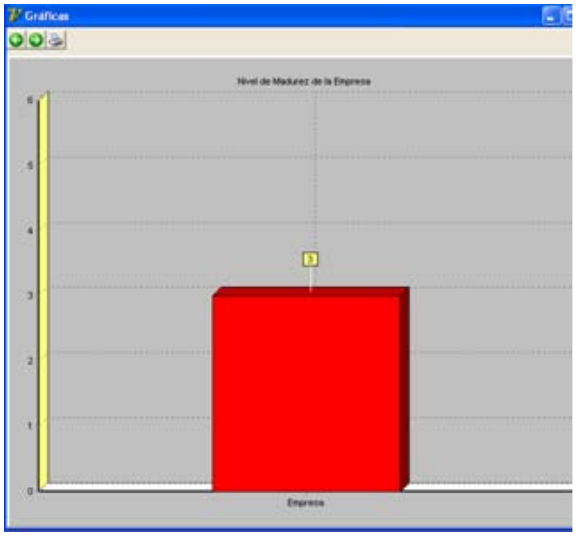


Figura 6: Niveles de madurez

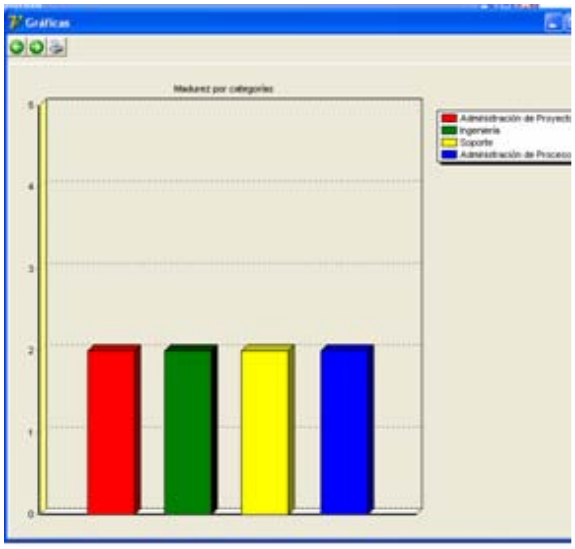


Figura 7: Madurez por categorías

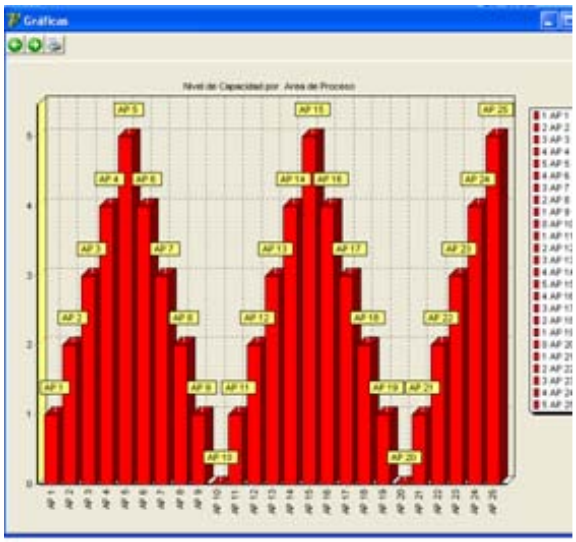


Figura 8: Niveles de capacidad por área de proceso

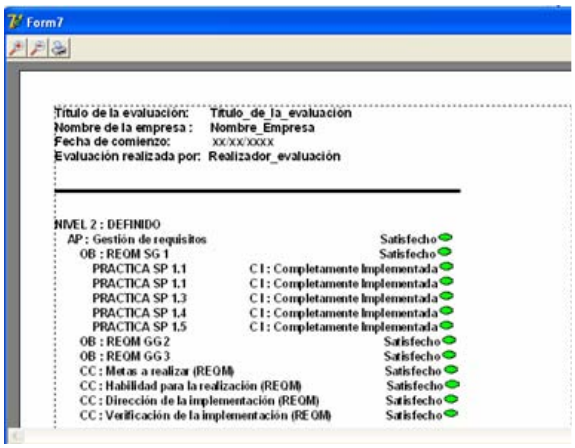


Figura 9: Informe del proyecto

Como complemento a las representaciones gráficas, pulsando sobre el botón Informes  de la interfaz principal, obtendremos un informe de resultados.

Además se ofrece la opción de abrir, editar, guardar y cerrar la evaluación en cualquier momento, facilitando de esta manera que la recolección de los datos se produzca al mismo tiempo que se usa el software.

Calidad a nivel de proyecto

En esta propuesta se trabaja con la metodología DUM y el objetivo de la aplicación es conocer si se está aplicando dicha metodología de forma adecuada, si los documentos generados son los correctos y si se generan en el orden adecuado.

Para ello, en primer lugar, el encargado de llevar a cabo la evaluación del proyecto deberá contestar una serie de preguntas efectuadas sobre cada tarea ejecutada. De esta manera, según las respuestas se podrá realizar una calificación a la tarea.

Dichas preguntas estarán basadas en el contenido de la documentación generada por cada tarea, de manera que podremos saber si los datos que contienen son los que deberían haberse generado con la ejecución de la misma.

Como resultado de este cuestionario sobre la documentación se obtiene una calificación de calidad sobre el proyecto. Además, una de sus principales ventajas es que no es necesario esperar a la finalización del proyecto para realizar una evaluación del mismo, pudiendo efectuarse en cualquier punto de desarrollo del proyecto.

El software mostrará en pantalla, por cada tarea, la documentación generada por la ejecución de la misma y el cuestionario a rellenar, de manera que con una simple lectura de la documentación se puedan contestar con facilidad.

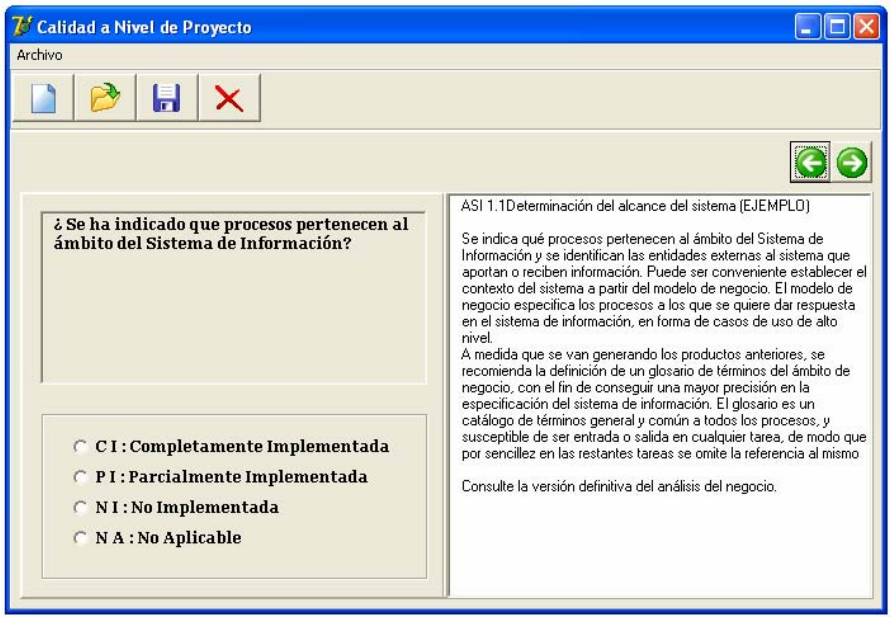


Figura 10: Formulario de evaluación de una tarea

Además de la evaluación explicada en el punto anterior, el software de manera automática realiza un análisis de los retrasos que se han sufrido con respecto a la planificación, estableciendo los puntos más débiles de los procesos en curso.

Como paso final en la evaluación, el software procederá a la generación del informe de resultados, obteniéndose una valoración de la calidad a nivel de proyecto.

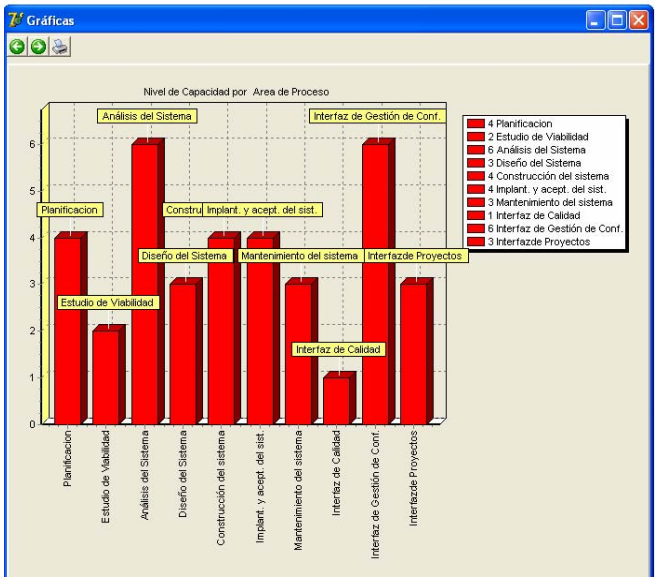


Figura 11: Resultados de la evaluación de las distintas líneas de trabajo

Cada una de las evaluaciones realizadas podrá ser guardada para una posterior actualización conforme se continúe con el desarrollo del producto, o editada ante posibles modificaciones.

Evolución de la calidad

Para poder establecer criterios de evolución de calidad, el sistema lleva un registro histórico de proyectos. Cada proyecto finalizado es almacenado en una base de datos para ser utilizado como referente para obtener un nivel de calidad de la empresa, y poder establecer puntos críticos de evolución para determinar la progresión de la empresa.

El tener disponible esta información en tiempo de desarrollo brinda la posibilidad de realizar comparativas entre la calidad de los proyectos en curso y el nivel actual de calidad de la empresa, permitiendo establecer controles para establecer si se mejora o empeora el nivel de calidad de la empresa

4 Conclusiones

Indudablemente, el concepto de calidad está estrechamente vinculado a la cultura u organización del trabajo. La gestión de la calidad a nivel de empresa u organización consiste en la creación de una estructura organizativa apropiada para fomentar el trabajo por la calidad de todas las personas y departamentos de la empresa. Por lo tanto, las empresas que adopten metodologías o esquemas de trabajo para mejorar el proceso de desarrollo de software tendrán muchas más ventajas competitivas frente a las demás.

En este contexto, la aplicación desarrollada pretende apoyar el seguimiento y evaluación permanente de la calidad en el desarrollo del software de una manera sencilla, confiable y segura, permitiendo a las empresas realizar un control permanente de los proyectos en curso. Asimismo, permite evaluar la eficacia con que se aplican las distintas tareas que componen la metodología de desarrollo. El sistema clasifica los resultados obtenidos en

función de la línea de trabajo a la que pertenecen, permitiendo tener una medida fiable del cumplimiento de la planificación prevista para los proyectos.

Otra ventaja que provee la aplicación, está relacionada con la recolección automática de una serie de datos referentes a la seguridad del sistema: riesgo al que están sometidos los activos, riesgo umbral de los mismos, existencia de plan de contingencia por activo, etc. que también son empleados como indicadores del estado actual de la calidad de los proyectos en curso.

Agradecimientos

Este trabajo esta incluido en el proyecto TIN2006-14285. Ministerio de Educación y Ciencia. España.

Referencias

AENOR (1995), UNE-EN-ISO 8402 Gestión de la calidad y aseguramiento de calidad. Vocabulario (ISO 8402:1994).

CMMI (2007). Capability Maturity Model Integration. Software Engineering Institute. Carnegie Mellon University.

IEEE Standard 610 (1990). Institute of Electrical and Electronics Engineers Computer dictionary. Compilation of IEEE Standard Computer Glossaries.

ISO/IEC (2002). Software Engineering – Software quality – General overview, reference models and guide to Software Product Quality Requirements and Evaluation (SQuaRE). Reporte. JTC1/SC7/WG6.

Peláez, J.I., Gámez J.I. Doña, J.M., (2007). DUM: Desarrollo Unificado con Métrica. Manuales Universidad de Málaga.

Pressman, R (2004). Ingeniería del Software: Un enfoque Práctico. McGraw Hill.

SCAMPI (2001). Standard CMMI Appraisal Method for Process Improvement. Handbook, Carnegie Mellon Universtity.

Sommerville, I.(2005). Ingeniería de Software. 7a Edición. Addison Wesley Publishers Limited.

Un Marco de Trabajo para Analizar y Mejorar la Calidad de Datos dentro de su Ciclo de Vida[1]

Gonzalo Domingo
Proyectos de Telesupervisión y Geociencias
D.S.I. Cuenta E&P - Argentina Sur
Repsol YPF
gedomingoe@repsolypf.com
y
Agustina Buccella, Alejandra Cechich
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue, Neuquén, Argentina
{abuccell, acechich}@uncoma.edu.ar

Resumen En la actualidad pocas empresas en la Argentina tienen en cuenta a la calidad de datos como requisito fundamental en todo desarrollo, implementación y uso del sistema. Es muy común que al momento de diseñar la aplicación, la calidad de datos sea muchas veces obviada y no exista una metodología o técnicas para su análisis. En este trabajo se propone una metodología orientada a pensar los sistemas desde la óptica de la calidad de los datos durante todo el ciclo de vida de un desarrollo de software, desde el momento del relevamiento y hasta la puesta en producción. La metodología cuenta con una serie de prácticas a realizarse de manera de garantizar dentro de una empresa la calidad de los datos cuando el sistema este en funcionamiento. Ilustramos la propuesta con un caso de estudio.

Palabras Clave: Calidad de Datos, Ciclo de Vida del Dato, Ciclo de Vida del Desarrollo de Software

1. Introducción

El término *Calidad de Datos* posee varias definiciones en la literatura [7,2,6], pero todas convergen en que el concepto calidad del dato es relativo al uso del dato [7]. Esto implica que este concepto es relativo, datos considerados con calidad para cierto uso pueden considerarse con insuficiente calidad para otros usos. Siguiendo con la definición, frases como: "Basura adentro, basura afuera"; "si se ingresa información inexacta, se obtendría información inexacta"; "pagar ahora o pagar mas tarde más" son muy comunes dentro del ámbito de calidad del dato. En nuestro trabajo, tomamos la definición de calidad del modelo FUNDIBQ (Fundación Iberoamericana de la Calidad) que establece, *la calidad es el conjunto de características propias de un producto, servicio, sistema o proceso imprescindibles para cumplir las necesidades o expectativas de partes interesadas*, es decir que la calidad es un punto de acuerdo entre las partes interesadas.

Una baja calidad de datos dentro de una empresa o institución lleva por ejemplo a clientes insatisfechos cuando sus datos personales, sus pedidos o sus facturas no son correctas; a empleados insatisfechos ya que cometen errores o no conocen cierta información, lo que los hace cometer a su vez más errores; a toma de decisiones erróneas porque los datos usados por los gerentes también pueden tener errores y es sabido que las decisiones no van a ser mejores que los datos en los que están basadas; etc. Por lo tanto se busca aprovechar los beneficios de una buena calidad de datos que se ven reflejados en la empresa o empresas que hacen uso de los mismos datos. Por ejemplo, mejora en el soporte a la

[1] Este trabajo esta parcialmente soportado por el proyecto UNCOMA 04/E059 (Mejora del Proceso de Desarrollo de Software Basado en Componentes)

toma de decisiones, reducción del tiempo necesario para obtener un informe, sustitución de actividades de bajo valor por otras de mayor valor, mejora de la imagen de la empresa, etc.

Para la elaboración de nuestro trabajo, hemos analizado varias propuestas que actualmente existen en la literatura. Entre ellas podemos citar a [3, 4, 8] ya que poseen en algún punto similitudes a nuestra propuesta. La propuesta de Wang [8] define una metodología denominada *Administración de la Calidad del Dato Total* (TDQM - Total Data Quality Management) cuyo objetivo es generar productos de información de alta calidad para los consumidores de información. La metodología propone, luego de analizar y conceptualizar el producto de información, la construcción de sistemas que fabrican o manufacturan la información (SMI). Estos SMI detallan las funcionalidades del sistema con los controles de calidad que debería poseer. Es justamente aquí donde se identifican posibles problemas de calidad analizando cómo se producen los datos.

En la propuesta de Ken Orr [3] el concepto de calidad de datos se basa en el uso del mismo y se establecen seis reglas para la calidad del dato como, los datos que no son usados no se mantienen correctos por mucho tiempo, la calidad de los datos está en función de su uso, no de su obtención, la calidad de los datos no será mejor que su uso más riguroso, etc. Teniendo en cuenta las reglas se definen una serie de actividades que apuntan a la evaluación y el análisis de la calidad de los datos. La actividad de *auditoria* consiste en determinar que tan buenos son los datos hoy. El *rediseño* se refiere a volver sobre las aplicaciones que están funcionando, enfocándose sobre todo en aquellos datos que puedan resultar más críticos para los procesos de negocio soportados por la aplicación y analizando cuidadosamente el uso que se le está dando a estos datos. La actividad de *entrenamiento* se centra en hacer comprender a los usuarios la importancia de la calidad de los datos. Así, se dedica tiempo a educación y entrenamiento. Por último la actividad de *medición* se refiere a medir constantemente la calidad de los datos, es decir, todas las actividades anteriores deben repetirse en el tiempo, haciendo a este un proceso iterativo. En comparación con nuestro trabajo, la metodología que proponemos intenta definir una guía práctica, aplicable a todos los sistemas que se van a construir en la empresa. Las recomendaciones o prácticas que creamos serán luego calificadas y no se basarán en su uso, sino en cómo afectan a las dimensiones de calidad del dato. Sin embargo, muchas de las prácticas que recogimos e incluimos en el marco de trabajo fueron recopiladas pensando en las reglas definidas en esta propuesta.

Por último, la propuesta de Pierce [4] plantea medir la calidad de los productos de información emanados de los sistemas a través de múltiples dimensiones tales como la certeza, la accesibilidad, la consistencia, etc. Para esto se utilizan matrices de control para combinar problemas con controles de calidad y así evaluar los productos de información. Las columnas de la matriz enumeran los problemas de la calidad de los datos que pueden afectar el producto de información; y las filas de la matriz son los controles de la calidad ejercitados durante el proceso de fabricación de la información para prevenir, detectar, o corregir estos problemas de la calidad del dato. Así, estos controles ayudan a evitar que cierto error aparezca en el producto de información. Nuestro trabajo tomó de esta propuesta la idea de basar el concepto de calidad en el ciclo de vida del dato. También implementamos la evaluación de las aplicaciones utilizando una matriz. Luego, clasificamos las distintas recomendaciones de calidad del dato encuadrándolas en el ciclo de vida del mismo y realizamos el análisis para evaluar si se respetaban los parámetros que se establecieron.

Este trabajo está organizado como sigue: La Sección 2 muestra el marco en el que se desarrolló la metodología. La Sección 3 explica la metodología en sí. La Sección 4 ilustra la propuesta con un caso de estudio. Conclusiones y trabajo futuro se discuten en la última sección.

2. La Calidad del Dato

En Redman [6] se describe el ciclo de vida del dato, el cual se compone de cuatro etapas fundamentales: *modelado del dato*, *captura del valor*, *almacenamiento* y *visualización*. La Figura 1 muestra gráficamente estas cuatro etapas.

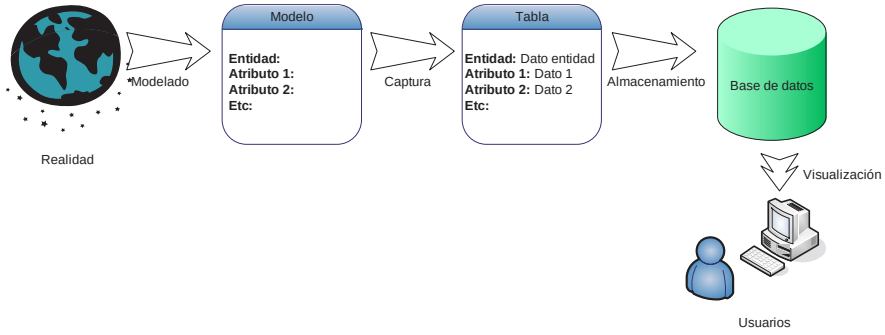


Figura 1. Ciclo de vida del dato.

La etapa de *modelado del dato*, se refiere a la elaboración de una abstracción representando la realidad una vez que han sido relevados los requerimientos o necesidades del cliente. Esta abstracción constituye un modelo lógico donde se establecen qué datos se tomaran y cómo fluirá la información por la aplicación y los roles de los actores que interactúan con ella. En la etapa de *captura del valor*, el dato es tomado de la realidad a través de interfaces del sistema o con otros sistemas. Aquí deben tenerse en cuenta aspectos como máscaras, fechas, validaciones y/o reglas en las interfaces de usuario. En la etapa de *almacenamiento*, el dato pasa de la interface de captura al repositorio de datos donde quedará almacenado. Por último la etapa de *visualización* se refiere a la presentación de los datos al usuario. Se deben tener en cuenta aspectos como la comprensión de la información, la identificación de errores o inconsistencias, la robustez, etc.

Para analizar y evaluar la calidad de los datos dentro de un proceso de desarrollo de software debemos tener en cuenta cuatro dimensiones principales [1]:

- *Exactitud*: ¿Representan los datos exactamente la realidad o fuentes verificables? La exactitud del dato esta relacionada con su fuente; es decir, es el nivel de correspondencia entre el dato y el mundo real.
- *Completitud*: ¿Todos los datos necesarios están presentes? ¿Qué cantidad de datos no están presentes? Esta dimensión se refiere a los datos necesarios que debe contener un sistema de información.
- *Consistencia*: ¿Los datos fueron consistentemente definidos y entendidos? Se refiere a la definición de estándares y protocolos para los datos. Todos los datos se representan en un formato compatible, que además es el más adecuado para la tarea que se está desarrollando.
- *Temporalidad*: ¿Los datos están disponibles cuando se necesitan? Por ejemplo, los datos ¿están disponibles cuando se deben tomar decisiones? Dentro de esta dimensión se enmarca el concepto de *Volatilidad*, el cual se refiere a la cantidad de tiempo que el dato se mantiene válido.

Estas cuatro dimensiones serán utilizadas para evaluar la calidad de datos en una empresa, ya que en ellas se enmarca la clasificación de las prácticas de acuerdo a la etapa del ciclo de vida del dato que afecten.

3. Un Enfoque para Mejorar la Calidad del Dato durante el Ciclo de Vida

Para la elaboración de nuestro nuevo enfoque, se relevaron las prácticas más utilizadas dentro de todo proceso de desarrollo de software y se analizaron de acuerdo a la calidad de los datos. Basados en las cuatro dimensiones de la calidad del dato, vistas en la sección anterior, clasificamos estas prácticas de acuerdo a la etapa del ciclo de vida del dato en que se encuentran. Definimos 64 prácticas en total – 27 prácticas para la etapa de modelado del dato, 22 para captura del valor, 4 en almacenamiento, y 11 en visualización. Por razones de brevedad no explicamos cada una de ellas, pero si daremos una breve descripción de algunas.

Ciclo de Vida del Dato: Modelado del Dato

- **Cuando se desactiva una cuenta de usuario se debe notificar, dependiendo del rol, al responsable del flujo sobre acciones preventivas**: De esta forma se mantiene actualizado el flujo de negocio ya que a veces serán necesarias acciones derivadas de este cambio. De esta forma, se afecta la *Temporalidad,* ya que el dato se mantendrá valido en el tiempo. Si cambia, se notifica a los responsables del dato (los consumidores de información definidos como referentes) detectando tempranamente diferencias con la realidad. La *exactitud* se ve afectada ya que permite detectar de forma temprana diferencias entre el dato almacenado y la realidad, minimizando el impacto negativo de la duplicidad de datos.
- **Si el dato existe en un sistema fuente, tomarlo de la misma**: Hay aplicaciones que por su dominio son consideradas fuentes de datos ya que son las que capturan el dato lo más cerca posible de su generación. Una vez que un sistema está definido como tal, si el dato es necesario para otra aplicación, no debe duplicarse, sino que debe existir una interface entre este y el sistema fuente. De esta forma no solo se garantiza unicidad sino que se mejora el uso e implementación del sistema fuente. Esto mejora la dimensión de *temporalidad* puesto que con esta práctica se favorece a que el dato se mantenga válido por más tiempo y que se detecten cambios tempranamente mejora la *consistencia* ya que al relacionar los sistemas se mantiene la definición del modelado del dato y se mejora la misma. La *completitud* mejora ya que se favorece que todos los elementos necesarios del dato estén presentes al interrelacionar los sistemas que hacen uso del mismo. La *exactitud* también mejora, al mejorar el uso e implementación del dato, la relación del mismo con la realidad se ve favorecida. Por otro lado al evitar la duplicación de datos, se disminuye la probabilidad de errores.
- **Evitar que un dato esté duplicado en más de un sistema**: Con esta práctica no solo se garantiza unicidad sino que se mejora el uso e implementación de los sistemas participantes. Con esto, se gana en *temporalidad* porque el dato se mantendrá válido por más tiempo y se podrán detectar cambios tempranamente y en *exactitud* al mejorar el uso e implementación del dato se favorece la relación del mismo con la realidad. Por otro lado al evitar la duplicación de datos, se disminuye la probabilidad de error.

Ciclo de Vida del Dato: Captura del Valor

- **Para la Codificación de las tablas tipificadoras, realizar consultas *like* antes de realizar una nueva inserción**: Este punto se refiere a permitir ingresar un dato luego de haber hecho una comprobación de si el mismo existe en la base de datos. Por ejemplo, si se ingresa una calle y se coloca como dato "Rivadavia" el sistema debiera consultar en la base de calles y comprobar que existen dos datos coincidentes. Por lo tanto se debe preguntar al usuario si se refiere a "Comodoro Rivadavia" o a "Bernardino Rivadavia". Así se mejora la *consistencia* ya que se asegura que el dato se mantiene consistente ya que no se guarda el ingreso sino la coincidencia con la tabla tipificada, la *completitud* guardando el dato completo y no solo lo que se ingresa y la *exactitud* disminuyendo la posibilidad de error de carga y eliminando la posibilidad de error de tipeo.
- **Si los datos a ingresar son críticos, evaluar el ingreso de los datos más de una vez**: Esto debe ser evaluado con el usuario referente para evitar que la carga sea tediosa. Por otro lado minimiza el error al combinar las probabilidades. Se mejora la *exactitud* al minimizar la probabilidad de error de tipeo.

Ciclo de vida del Dato: Almacenamiento

- **Si existe una regla matemática para inferir un campo a través de otro, este no se debe cargar**: Esta regla de inferencia debe estar modelada en la aplicación para evitar así el error de ingreso de datos. Así se puede mejorar la *temporalidad* al ayudar a mantener el dato válido porque al cambiar los datos que le dieron origen, estos se actualizarán; la *consistencia* ya que el dato tendrá el formato esperado porque se define dentro de la aplicación; la *completitud* debido a que no hay ingreso humano y la regla deberá validar que los datos que le dan origen lo hacen en toda su completitud y la *exactitud* ya que la correspondencia con la realidad se mantendrá mientras la regla de inferencia esté bien modelada.
- **Las reglas de negocio relevadas deben ser parte de la aplicación para que el dato sea almacenado y filtrado por estas reglas. El mismo debe ser dinámico**: Si se filtra la información por las reglas de negocio se puede detectar tempranamente falta de correspondencia entre el dato ingresado y lo esperado, evitando así el ingreso de datos basura. De esta forma mejora la *temporalidad* ya que el dato que ingresa es más estable y tiende a no quedar desactualizado; la *consistencia* ya que se ayuda a que el dato ingrese en el formato esperado y la *exactitud* porque se minimizan errores de ingreso de datos.

Ciclo de Vida del Dato: Visualización

- **El sistema debe alertar sobre vencimientos**: De esta forma el responsable de los datos, que es el usuario referente de la aplicación o quién se haya designado, es avisado cuando de acuerdo a la lógica de la aplicación algún dato está por perder validez. Mejora así la *temporalidad* ayudando al responsable a tomar acciones preventivas y la *exactitud* ya que al perder validez el dato se vuelve inexacto, por lo cual si esto se sabe con anticipación y se re valida, el riesgo de pérdida de correspondencia con la realidad disminuye.
- **El sistema debe verificar y advertir cambios en la tendencia de los datos**: De esta forma, se puede advertir en modo preventivo un cambio de tendencia. Para determinar si se trata de un error o de un cambio efectivo en la tendencia de la realidad, se requerirá un análisis funcional que

deberá realizar el dueño de los datos. Puede servir para detectar tempranamente errores en el registro del dato. Mejora así la *exactitud* al alertar de desvíos para tomar acciones preventivas destinadas a localizar falta de calidad en el registro del dato.

- **El proceso de negocio soportado tiene que estar abierto a otros procesos (cultura de compartir los datos)**: Esta práctica es más de negocio que de sistemas. Pero es una recomendación que debemos realizar al negocio cuando estamos observando un proceso que debiera alimentar o alimentarse de otros procesos de la compañía pero no lo hace. Se espera que de esta forma mejore la calidad, intensificando el uso de los datos.

La clasificación de las prácticas está afectada por dos criterios según la independencia del sujeto que realiza el análisis, *Objetivo* y *Subjetivo*. El primero de ellos se realiza de acuerdo a un comité de calidad que evalúa las prácticas y el segundo se aplica el criterio del evaluador que podrá variar de acuerdo a su conocimiento del dominio y a su experiencia previa. Ambos determinan mediante una tabla de puntaje un valor de error si la práctica no se cumple.

La Clasificación Objetiva se enumera a continuación:

- *Práctica Estándar (E)*: La aplicación de este punto es considerada un estándar en la industria y su aplicación debería masificarse.
- *Buena práctica (B)*: Los puntos calificados de esta forma deberían aplicarse siempre que sea posible. Se considera que son muy importantes para garantizar que los datos sean de calidad en las dimensiones que afectan, pero se contempla que a veces resultan difíciles de aplicar por su costo, siendo antieconómicas. En estos casos, se debe documentar la decisión de no aplicarlas como parte del diseño.
- *Recomendación (R)*: La aplicación de estos puntos se considera favorable, quedando a criterio del líder del proyecto la evaluación costo/beneficio para su aplicación.

La Clasificación Subjetiva posee los siguientes valores:

- *Sin Error*: El soporte funcional observa que la práctica en cuestión está aplicada de forma correcta en la aplicación.
- *No Aplica*: La práctica recomendada no se observa, pero no se considera error ya que esta decisión de no aplicarla fue tomada en tiempo de diseño y la misma está documentada.
- *Leves*: Situaciones que pueden disminuir la calidad del dato. Cuando se permite el ingreso de datos de baja calidad pero no afecta a los datos que son críticos ni al éxito de la tarea que esta realizando el usuario.
- *Graves*: Situaciones propensas a disminuir la calidad del dato que pueden afectar el éxito de la tarea. Cuando la aplicación permite que se ingrese un dato que al ser erróneo pueda comprometer la tarea que se está realizando.
- *Fatales*: Errores conceptuales, aplicación de un modelo erróneo o errores que impiden terminar la tarea exitosamente. Son los más peligrosos ya que permiten que se ingresen datos que impiden terminar la tarea para la cual se los está capturando.

3.1. Mejora al Proceso de Desarrollo

Nuestro trabajo fue desarrollado en una empresa del medio que por confidencialidad, llamaremos “El Petróleo SA”. En dicha empresa, el desarrollo de sistemas, desde la obtención de

requerimientos hasta la puesta en producción del mismo, estaba guiado por un proceso que no consideraba la tarea de verificar y controlar la calidad de los datos como requisito temprano. Luego del análisis exhaustivo de calidad de datos realizado por dicha empresa, en donde fueron definidas las prácticas y recomendaciones antes mencionadas, el proceso de desarrollo fue modificado. La Figura 2 muestra la parte de este proceso con los controles de calidad agregados.

En la figura se observa cómo interactúan los actores que intervienen en el desarrollo de una aplicación. Estos actores son: *Usuario Referente*, *Analista Funcional*, *Proveedor* y *Soporte Funcional*.

En las etapas de relevamiento y elicitación de requerimientos, el usuario referente aporta las historias denotando su forma de representar los requerimientos funcionales del sistema. Por otro lado, el analista funcional determina las pautas de calidad. Este trabajo se realiza junto con el usuario para llegar a un acuerdo de cuáles de las prácticas definidas (evaluando el peso y costo/beneficio de cada una) serán puestas en práctica en la aplicación. Este trabajo genera un documento con las pautas de calidad acordadas que se entrega al proveedor de aplicación, sea este el equipo de desarrollo o el proveedor de aplicación en caso de productos que ya están desarrollados.

El equipo de proyectos se pregunta (no mostrado en la figura por cuestiones de espacio) , en esta instancia, si conoce una solución corporativa que aplique a la necesidad. Si es así, se implementa con consenso de la gerencia de proyectos. Si no se conoce una solución preestablecida, se elabora el documento de visión que enumera las necesidades del negocio, el objetivo de una solución informatizada y su alcance.

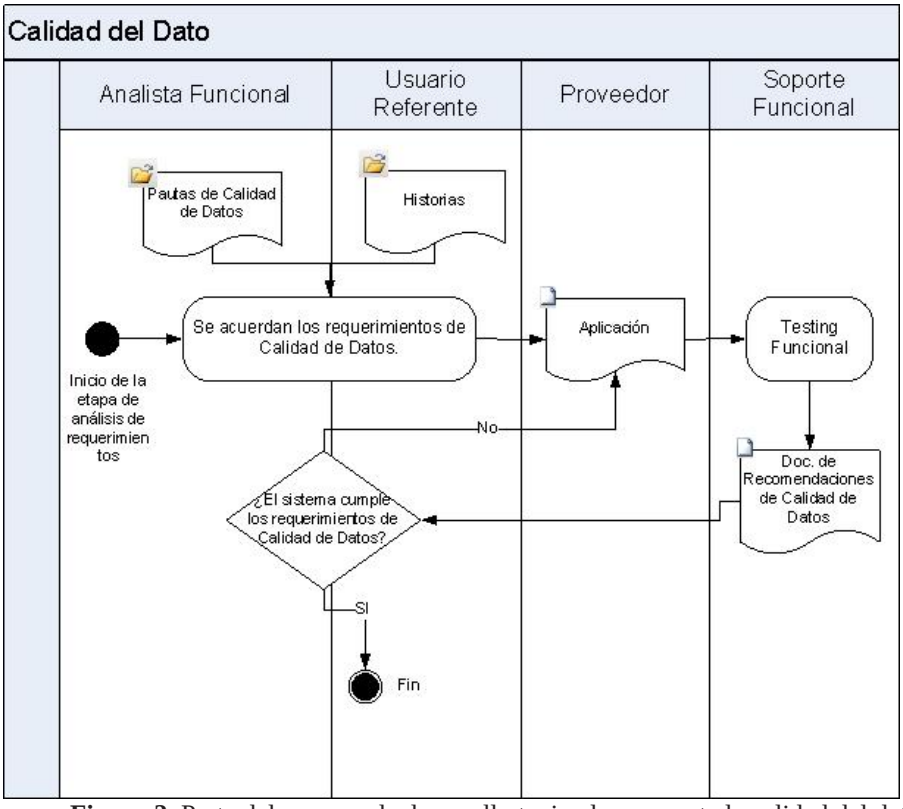


Figura 2. Parte del proceso de desarrollo teniendo en cuenta la calidad del dato.

Si el desarrollo será local, se empieza con la etapa de relevamiento detallado y diseño. El documento de historias se pasa al grupo de programación para que estime esfuerzo de desarrollo en

tiempo por historia. Con esto, el grupo de proyectos arma el cronograma y el plan de entregables. El mismo se valida con el usuario para ajustar prioridades y negociar tiempos de entrega teniendo como variable de ajuste el alcance de cada entrega. De estas entrevistas surgen como documentos el cronograma y el plan de entrega que detalla cada entregable con su alcance.

Se agregan a la documentación que se entregará al grupo de programación las pautas de calidad de datos. Estas son las 64 prácticas excepto que alguna se desestime por una decisión de diseño.

Luego, avanzando con el cronograma, se devuelve el código fuente de la aplicación, la aplicación funcionando sin errores unitarios ni de integración[2] y el manual de instalación continuando con la programación de la siguiente entrega. El grupo de proyectos recibe dicha entrega y la instala en un servidor de prueba donde realiza las pruebas funcionales generando documentos de errores y mejoras detectadas. A su vez, el Soporte Funcional utiliza estas pruebas funcionales como guía para realizar las pruebas de calidad de datos. Aplica el marco de trabajo para obtener así los valores de error y la recomendación de aprobación o no de la prueba.

En base a los resultados obtenidos, el Soporte Funcional, elaborará también el documento de recomendaciones de calidad del dato, donde dejará explicitada la descripción de las dimensiones que presentan errores, comentando todos los puntos en los que considere que no se respetaron las pautas o no se cumplen las recomendaciones y el motivo por el cual no encuadra con lo esperado. También se detallan las acciones para acceder o visualizar el dato en cuestión.

Con el documento de recomendaciones de calidad del dato, entre el analista y el usuario referente, decidirán si la aplicación cumple con un mínimo de pautas para entrar a producción o si se deben re-enviar los resultados al proveedor para que mejore el producto antes de iniciar su ciclo productivo. Si aparecen correcciones o ajustes tanto de calidad como funcionales, se re planifica y se devuelve a desarrollo. Si el usuario referente da su aprobación, pasa a un ambiente de producción donde lo pueden acceder todos los usuarios.

3.2. Aplicación de nuestro Marco de Trabajo: Verificando Aplicaciones

En nuestro marco de trabajo, cada práctica está ponderada de acuerdo a cada dimensión de la calidad del dato. La escala se puede ver en la Figura 3.

Ponderación	
Estándar	3
Buena práctica	2
Recomendación	1
No aplica	0

Figura 3. Tabla de ponderación de las prácticas (Clasificación Objetiva).

Esta ponderación objetiva se combina con la apreciación del soporte funcional. Éste, en el momento de la verificación, evalúa la aplicación de acuerdo al cumplimiento de cada práctica. Esta evaluación se efectúa de acuerdo a la escala de la Figura 4.

[2] En programación, una prueba unitaria es una forma de probar la corrección de un módulo de código, esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego con las Pruebas de Integración se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión

Luego, el valor de error se multiplica para cada práctica y cada dimensión por el de ponderación dando el valor de incumplimiento de cada práctica. Finalmente se suman los valores de todas las prácticas para determinar el valor final. De esta manera, si el valor de error es menor a 46, se recomienda aprobar la prueba. Si está entre 46 y 119 se recomienda aprobarla con observaciones y si el valor de error es mayor a 120, la recomendación de la prueba será rechazar la aplicación.

Estos valores fueron escogidos para que si una práctica estándar tiene un error fatal, (equivale a 40 x 3) se genere un valor de error de rechazo, y así cualquier valor de error mayor dará una recomendación de rechazo también. La aprobación se dará con cualquier valor de error menor o igual a 45. Esto se estableció considerando que un error grave en una práctica estándar era lo máximo aceptable para aprobar. La aprobación con observaciones será cualquier combinación de valores intermedios.

Error	Puntos	Descripción
Sin Error	0	
No Aplica	0	
Leves	5	Situaciones que pueden disminuir la calidad del dato
Graves	15	Situaciones propensas a disminuir la calidad del dato que pueden afectar el éxito de la tarea
Fatales	40	Errores conceptuales, aplicación de un modelo erróneo o errores que impiden terminar la tarea exitosamente.

Figura 4. Tabla de errores de las prácticas (Clasificación Subjetiva).

Los resultados serán validados con el usuario referente ya que el rechazo implica volver a desarrollar y planificar las entregas del producto. Será una decisión consensuada ya que existe la posibilidad de que el usuario referente igualmente apruebe el pasaje a producción de la entrega y que los errores encontrados se solucionen en la siguiente etapa.

Todos los errores encontrados y las decisiones tomadas en base a éstos quedarán documentados en el Documento de Recomendaciones de Calidad de Datos (visto en la Figura 2). En el mismo se detallan las recomendaciones que no se cumplen, cómo afectan a cada dimensión describiendo el error y el motivo por el cual no encuadra con la dimensión en cuestión.

4. Un Caso de Estudio

Como caso de estudio se utilizó una entrega de una aplicación que administra la electricidad de la compañía “El Petróleo SA”. Esta aplicación centraliza la información acerca de la electricidad generada por la compañía, la electricidad comprada y registra las ventas de energía eléctrica, como así también los datos de los equipos generadores de electricidad. La disponibilidad de esta información tiene, entre otros objetos, la generación de informes solicitados por la Secretaría de Energía de la Nación.

Antes de una solución informática, la información era mantenida en planillas, lo que dificultaba la generación de los informes a la Secretaría de Energía y otros entes reguladores. La generación manual de estos informes era compleja ya que había que corroborar grandes cantidades de datos para evitar inconsistencias. Los datos de la energía generada eran mantenidos localmente en cada planta y centralizados en forma manual por personal del área de energía eléctrica.

Para el desarrollo de este módulo y como era la primera vez que se aplicaba, se reunió a los desarrolladores, el analista, el soporte funcional de la aplicación y se los instruyo en los roles que cada

uno asumiría para esta prueba del proceso modificado y orientado a la calidad del dato en las aplicaciones.

Antes de comenzar la programación de la entrega que se tomó para la prueba, se les entregó a los desarrolladores las 64 prácticas definidas, junto con la categorización de las mismas de acuerdo a las cuatro dimensiones del dato y agrupadas de acuerdo a su ciclo de vida.

Cuando el equipo de desarrollo superó las pruebas unitarias y las pruebas de integración para asegurar el correcto funcionamiento del sistema o subsistema en cuestión, se le aplicó al entregable una prueba funcional. En ese momento el soporte funcional tomó la aplicación y ejecutó la prueba funcional, la prueba de usabilidad y el de calidad del dato, de acuerdo a la modificación implementada en el proceso de desarrollo.

Para esto, el soporte funcional, revisó cada una de las recomendaciones y verificó su cumplimiento, registrando si detectaba alguna falta o falla.

A continuación se explicarán algunas de las observaciones encontradas tal y como se documentaron en el "Documento de Recomendaciones de Calidad de Datos".

Ciclo de Vida del Dato: Modelado del Dato

- **Cuando se desactiva una cuenta de usuario se debe notificar, dependiendo del rol, al responsable del flujo sobre acciones preventivas**: En la dimensión temporalidad se consideró un error "Leve", ya que no se controlaba. Igualmente, la cantidad de usuarios en esta etapa del ciclo de vida de la aplicación no justificaba que se considere como un error de mayor envergadura. En exactitud se colocó "No aplica" por la razón antes explicada. De esta forma quedó documentado para ser tenido en cuenta en las siguientes entregas del desarrollo.
- **Si el dato existe en un sistema fuente, tomarlo de la misma**: En la aplicación existen datos que están en sistemas fuentes como los contratos de comercialización que se llevan con SAP. El sistema tiene una interface con éste por lo que se pudo verificar que no hay error.
- **Evitar que un dato esté duplicado en más de un sistema**: Se puso un valor de error "Leve" en temporalidad y "No aplica" en exactitud. Ya que por disposiciones de seguridad informática los datos tomados del sistema fuente SAP, con el cual se conecta esta aplicación, no pueden ser accedidos en línea sino que se exportan una vez al día a los sistemas satélites. Por lo cual la temporalidad puede hacer que el dato no sea valido como máximo por 24 horas. Pese a ello para este desarrollo se acordó que no afectaría a la exactitud ya que los datos importados son poco dinámicos.

Ciclo de Vida del Dato: Captura del Valor

- **Para la Codificación de las tablas tipificadoras, realizar consultas like antes de realizar una nueva inserción**: Se consideró un error "Leve" para la exactitud y "No aplica" para consistencia y completitud ya que las mismas no se ven afectadas porque la redundancia está controlada por la base de datos.
- **Si los datos a ingresar son críticos, evaluar el ingreso de los datos más de una vez**: En esta práctica se puso un valor de "No aplica" en la dimensión exactitud, ya que hay circuitos y cadenas de aprobación definidos para los datos más críticos. Como por ejemplo, cuando se carga el valor de una factura de distribuidor de un determinado proveedor de energía eléctrica, se dispara un mail a la gente de "Cuentas a Pagar" quienes con una copia de la factura verifican que el valor sea correcto y aprueban el pago en la aplicación.

Ciclo de vida del Dato: Almacenamiento

- **Si existe una regla matemática para inferir un campo a través de otro, este no se debe cargar**: Se verificó y se consideró sin error.
- **Las reglas de negocio relevadas deben ser parte de la aplicación para que el dato sea almacenado filtrado por estas reglas. El mismo debe ser dinámico**: Se verificó y se consideró sin error.

Ciclo de Vida del Dato: Visualización

- **El sistema debe alertar sobre vencimientos**: Se verificó y se consideró sin error. Por ejemplo, una vez cargada la factura de distribución, el personal de "Cuentas a Pagar" tiene 5 días hábiles para pagar, la aplicación verifica este vencimiento y va alertando del mismo.
- **El sistema debe verificar y advertir cambios en la tendencia de los datos**: Se colocó error "Leve" en exactitud ya que está en desarrollo y se espera implementarlo en una etapa más avanzada de la implementación.
- **El proceso de negocio soportado tiene que estar abierto a otros procesos (cultura de compartir los datos)**: Se verificó y se consideró sin error. De hecho el usuario referente es partidario de la sinergia entre áreas. Valoró que Sistemas de Información tenga entre sus recomendaciones el verificar que esto se cumpla. Esta aplicación que es del área "Ingeniería de Petróleo Gas y Electricidad" será también usada por el área "Cuentas a Pagar" gracias a esta cultura de procesos abiertos.

Estos son algunos de los puntos que se analizaron y documentaron. La valoración final de puntos de error dio un valor de 90 puntos compuestos por: 15 puntos por una práctica estándar con error leve, 10 puntos por una buena práctica con error leve, 10 puntos por una buena práctica con error leve, 5 puntos por una recomendación con error leve, 15 puntos por una práctica estándar con error leve, 10 puntos por una buena práctica con error leve, 15 puntos por una práctica estándar con error leve, 10 puntos por una buena práctica con error leve. Luego de comparar el peso de 90 con la tabla antes explicada, la recomendación final fue: "Aprobar con observaciones".

Con esta recomendación, se decidió pasar a producción la aplicación, documentar estas observaciones, analizarlas y mejorar en la siguiente entrega modular de la aplicación. Con lo que las observaciones encontradas se solucionaron en la siguiente iteración del proceso, sin demorar la planificación de la entrega del módulo analizado.

5. Conclusiones y Trabajo Futuro

En este trabajo hemos partido de la definición de calidad de los datos como un punto de acuerdo entre las partes interesadas, es decir, las características que un producto debe cumplir para satisfacer las expectativas de los interesados. Es sabido, que los datos de las organizaciones son propensos a dejar de satisfacer las necesidades de dichas partes rápidamente.

Para dar soluciones a este problema, hemos creado un marco de trabajo el cual ha generado la modificación de nuestro ciclo de vida en el proceso de desarrollo de nuestros sistemas. Este marco

funciona como una guía de recomendaciones a aplicar en los sistemas que se desarrollan, permitiendo evaluar los mismos de acuerdo a la forma en que están construidos y haciendo énfasis en la calidad de los datos que manipularán.

La metodología descripta debe ser parte integral de una organización, del grupo de desarrollo y de la mentalidad de sus componentes. Para lo cual es necesario mejorar los procesos y la manera de trabajar; dar a la planificación el lugar que se merece y producir un cambio de cultura. Es necesario persuadir a los empresarios de que los beneficios de medidas preventivas son tangibles. La calidad de los datos no debe ser un agregado a las aplicaciones, sino algo que surja desde el propio diseño.

De cara al futuro nos queda, implementar este proceso en toda la cultura de la organización para que los sistemas sean desarrollados atendiendo a factores como la calidad del dato y decidiendo de este modo invertir ahora y no pagar más adelante el costo de la falta de calidad. De esta manera el marco será probado en varios sistemas y podremos medir su eficacia y proponer en casos de que se necesario nuevos cambios.

Referencias

[1] G. Brackstone**.** Managing data quality in a statistical agency. Survey Methodology, (25):139-179, 1999.

[2] E. M. Burns, O. MacDonald, and A. Champaneri**.** Data quality assessment methodology: A framework. In Joint Statistical Meetings Section on Government Statistics, pages 334-337, 2000.

[3] K. Orr. Data quality and systems theory. - Communications of the ACM, 41(2):66-71, February 1998.

[4] E. Pierce. Assessing data quality with control matrices. Communications of the ACM, 47(2):82-86, February 2004.

[5] T. Redman. The impact of poor data quality on the typical enterprise. - Communications of the ACM, 41(2):79-83, February 1998.

[6] T. Redman. Data Quality: The Field Guide. Digital Press, January 15 2001.

[7] G. Tayi and D. Ballou. Examining data quality. - Communications of the ACM, 41(2):54-57, February 1998.

[8] R.Wang. A product perspective on total data quality management. - Communications of the ACM, 41(2):58{65, February 1998.

Un nuevo índice Métrico-Temporal: el Historical FHQT

Anabella De Battista, Andrés Pascal
Univ. Tec. Nacional, Fac. Reg. C. del Uruguay, Dpto. Sistemas de Información
Concepción del Uruguay, Argentina, 3260
{debattistaa,pascala}@frcu.utn.edu.ar

Gilberto Gutiérrez
Universidad del Bío-Bío, Facultad de Ciencias Empresariales
Chillán, Chile, 3810563
ggutierr@ubiobio.cl

Norma Herrera
Univ. Nac. de San Luis, Departamento de Informática
San Luis, Argentina, 5700
nherrera@unsl.edu.ar

Abstract

Recently a new database model, the metric-temporal databases, has been proposed. This model uses concepts of metric spaces to efficiently solve similarity queries, and concepts of temporal databases to allow store and efficiently retrieve data with a temporal component. This new model combines both aspects to solve problems where is necesary perform similarity searches, but having in account the temporal component. For it, we present the Historical FHQT, a new metric-temporal index that uses several instances of metric structure FHQT in order to represent the alive objects in each moment, and then we verified experimentally the efficiency of this method for a determined set of queries.

Keywords: metric-temporal query, metric space, temporal database, metric-temporal database, Historical-FHQT

Resumen

Recientemente ha sido propuesto un nuevo modelo de bases de datos, las bases de datos métrico-temporales, que utiliza conceptos de espacios métricos para realizar eficientemente consultas por similitud, y de bases de datos temporales para permitir almacenar y recuperar eficientemente datos que poseen una componente temporal. Este nuevo modelo combina ambos aspectos con el fin de resolver problemas donde resulta de interés realizar búsquedas por similitud pero teniendo en cuenta también la componente temporal. En este artículo se propone el Historical FHQT, un nuevo índice métrico-temporal que utiliza varias instancias de la estructura métrica FHQT para representar los objetos vigentes en cada instante de tiempo, y luego verificamos su eficiencia experimentalmente para un conjunto determinado de consultas.

Palabras clave: consulta métrico-temporal, espacio métrico, base de datos temporal, base de datos métrico-temporal, Historical FHQT

1 Introducción

En la actualidad las bases de datos no sólo almacenan datos estructurados, sino también objetos tales como imágenes, sonido, texto, video, datos geométricos y otros tipos de datos que no pueden ser consultados mediante búsquedas exactas. A su vez, en algunas situaciones es necesario tener en cuenta el aspecto temporal, es decir, registrar la evolución de los datos a través del tiempo.

Para poder representar estos nuevos tipos de bases de datos han surgido modelos que permiten almacenar eficientemente esta clase de datos y realizar búsquedas eficientemente sobre los mismos. Los modelos sobre los cuales trabajamos en este artículo son las Bases de Datos Temporales, los Espacios Métricos y las Bases de Datos Métrico-Temporales.

- Las ***bases de datos temporales*** [14, 11] permiten almacenar y recuperar datos que dependen del tiempo. Mientras que las bases de datos tradicionales tratan al tiempo como otro tipo de dato más, este tipo de base de datos incorpora al tiempo como una dimensión. Por ejemplo, una consulta de interés en una base de datos temporal de catastro podría ser: "*conocer un instante o período de tiempo en que una parcela fue propiedad de cierta persona*".

- Los ***espacios métricos*** [7, 4, 5] son un modelo de bases de datos que permiten tratar con búsquedas por similitud, es decir, búsquedas de objetos parecidos o similares a uno dado. Este tipo de búsqueda tiene una amplia gama de aplicaciones, por ejemplo: reconocimiento de imágenes y sonido, compresión de texto, biología computacional, inteligencia artificial y minería de datos, entre otras [9, 13].

- Las ***bases de datos métrico-temporales*** [15] permiten realizar consultas por similitud y teniendo en cuenta el aspecto temporal: por instante, o por intervalo de tiempo. Un ejemplo de aplicación de este modelo, es el siguiente: supongamos una base de datos de imágenes de la policía federal donde se registran rostros de un grupo de individuos. Sobre esa base de datos sería de interés, dada la especificación de un rostro, buscar las personas que tenían rasgos similares en un momento determinado. Esta consulta implica buscar teniendo en cuenta tanto la componente métrica como la componente temporal.

En este artículo presentamos un nuevo método de acceso métrico-temporal orientado a las consultas por similitud sobre una base de datos de objetos no estructurados instantáneos, es decir, que sólo tienen vigencia en un instante de tiempo.

Este artículo está organizado de la siguiente manera. En la Sección 2 presentamos una breve reseña del trabajo relacionado. En la Sección 3 presentamos el HFHQT, un índice para resolver consultas métrico-temporales. En la Sección 4 mostramos los resultados de la evaluación experimental de este nuevo método de acceso y en la Sección 6 exponemos las conclusiones y el trabajo futuro.

2 Trabajo relacionado

En esta sección se presenta un breve resumen de los modelos de bases de datos y métodos de acceso a los que se hace referencia en este trabajo: las Bases de Datos Temporales, los Espacios Métricos y las Bases de Datos Métrico-Temporales.

2.1 Bases de datos Temporales

Las bases de datos temporales mantienen información acerca del pasado, el presente y en algunos casos, pueden predecir el futuro más probable. La dimensión temporal es manejada internamente por el sistema administrador de la base de datos. Una verdadera base de datos temporal es aquella que soporta *tiempo válido* y *tiempo transaccional*. El *tiempo válido* expresa el tiempo durante el cual una proposición es cierta. El *tiempo transaccional* indica el tiempo en el que una proposición aparece reflejada en la base de datos como cierta, es decir, el momento en que se incorpora esa información en la base de datos.

Existen tres clases de bases de datos temporales, en función de lo anterior:

- **De tiempo transaccional** (transaction time): registran el tiempo de acuerdo al momento en que se almacena un hecho, es decir, de acuerdo al orden en que se procesan las transacciones. Hay que notar, que este registro no necesariamente coincide con el orden real en que se produjeron los eventos, más bien, es acorde al tiempo en que la base tomó conocimiento del evento. Debido a que se mantiene la historia de todos los estados consistentes de la base de datos, se puede realizar un "*rollback*" hacia cualquiera de estos estados anteriores. Este tipo de bases de datos no permite modificar el pasado.

- **De tiempo válido o vigente** (valid time): soportan el tiempo en que el hecho ocurrió en la realidad, que puede no coincidir con el momento de su registro. Este sistema permite realizar correcciones sobre los datos registrados. En dicho caso, sólo se mantiene la última versión de cada estado.

- **Bitemporales**: integran la dimensión transaccional y la dimensión vigente a través del versionado de los estados, es decir, cada estado se puede modificar para actualizar el conocimiento de la realidad pasada, presente o futura, pero esas modificaciones se realizan generando nuevas versiones de los mismos estados.

Los tipos básicos de consultas a sistemas de bases de datos temporales son tres:

(1) dado un intervalo continuo, encontrar todos los objetos "vigentes" en ese período
(2) dados un rango de claves y un intervalo continuo, encontrar todos los objetos cuyas claves forman parte del rango, y que estuvieron "vigentes" dentro de ese período
(3) dado un rango de claves, devolver la historia de todos los objetos cuyas claves están en ese rango.

Existen casos especiales de estos tipos de consultas como por ejemplo, que los intervalos se reduzcan a un instante (*range-time slice*), o que el rango de claves contenga una sola clave (*pure-key* query) [11, 12, 14, 17].

2.2 Espacios Métricos

Las aplicaciones en las que se requiere realizar consultas por similitud tienen un marco conceptual común que da soporte a este tipo de búsquedas bajo alguna función de distancia o similitud determinada. En esta sección introducimos el modelo formal correspondiente.

Un espacio métrico es un par *(U, d)* donde *U* es un universo de objetos y $d : U \times U \rightarrow R^+$ es una función de distancia definida entre los elementos de *U* que mide la similitud entre ellos; esto significa que a menor distancia más cercanos o similares son los objetos. Esta función *d* cumple con las propiedades características de una función de distancia:

(a) $\forall x, y \in U, d(x, y) \geq 0$ (positividad)
(b) $\forall x, y \in U, d(x, y) = d(y; x)$ (simetría)
(c) $\forall x \in U, d(x, x) = 0$ (reflexividad)
(d) $\forall x, y, z \in U, d(x, y) \leq d(x, z) + d(z, y)$ (desigualdad triangular)

La base de datos será un subconjunto finito $X \subseteq U$ de cardinalidad *n*. En este nuevo modelo de bases de datos, una de las consultas típicas que implica recuperar objetos similares es la búsqueda por rango, que denotaremos con $(q, r)_d$. Dado un elemento $q \in U$ al que llamaremos *query,* y un radio de tolerancia *r*, una búsqueda por rango consiste en recuperar los objetos de la base de datos que estén a distancia a lo sumo *r* de *q*, es decir:

$$(q, r)_d = \{x \in X / d(q, x) \leq r \}$$

Para resolver esta clase de consultas con mayor eficiencia que *O(n)* evaluaciones de distancias (correspondiente a recorrer secuencialmente la base de datos), se utilizan estructuras auxiliares (índices) que permiten ahorrar cálculos durante el proceso de búsqueda.

Existen distintos enfoques para la construcción de algoritmos de indexación en espacios métricos; en este trabajo utilizamos la estrategia basada en Pivotes.

Los Algoritmos Basados en Pivotes trabajan de la siguiente manera: durante la indexación seleccionan *k* pivotes $\{p_1, p_2, ..., p_k\}$ y le asignan a cada elemento *a* de la base de datos el vector o firma $\Phi(a)=(d(a, p_1), d(a, p_2), ..., d(a, p_k))$. Durante la búsqueda usan la desigualdad triangular junto con la firma de cada elemento para filtrar objetos de la base de datos sin medir su distancia a *q*. Dada $(q, r)_d$, se computa la firma de la query *q*, $\Phi(q)=(d(q, p_1), d(q, p_2), ..., d(q, p_k))$, y luego se descartan todos aquellos elementos *a* tales que para algún pivote p_i, $|d(q, p_i) - d(a, p_i)|>r$. Los elementos no descartados forman parte de una lista de candidatos, que posteriormente se comparan directamente con la query *q*.

De los métodos basados en pivotes, se seleccionó el FHQT (Fixed Height FQT) [1] como base para el diseño de un índice métrico-temporal. Este índice pertenece al grupo de algoritmos basados en pivotes y admite dinamismo, una característica que no todos los índices métricos poseen. A continuación se explica el funcionamiento de este índice.

Fixed-Height FQT

En [1, 2] se presenta el *Fixed-Height FQT* o *FHQT* (Figura 1), que es una variante del Fixed Queries Tree (FQT) [1] en donde todas las hojas se encuentran a la misma altura. Originalmente estas estructuras fueron propuestas para funciones de distancias discretas, pero se pueden adaptar a distancias continuas discretizando los valores de las mismas [16, 8].

El árbol se construye a partir de un elemento *p* (pivote) que puede ser elegido arbitrariamente, o mediante algún procedimiento de selección de pivotes [6], del universo *U*. Para cada distancia *i* se crea el conjunto C_i formado por todos aquellos elementos de la base de datos que están a distancia *i* de *p*. Luego, para cada C_i no vacío se crea un hijo del nodo correspondiente a *p,* con rótulo *i,* y se construye recursivamente un FHQT teniendo en cuenta que todos los subárboles del mismo nivel usarán el mismo pivote como raíz. Este proceso recursivo, en el caso del FQT se repite hasta que

queden menos de *b* elementos, los cuales se almacenan en una hoja. En el caso del FHQT se continúa hasta lograr que todas las hojas tengan menos de *b* elementos y estén en un mismo nivel.
La Figura 1 muestra un ejemplo de un FHQT con dos pivotes. Ante una consulta $(q, r)_d$, se comienza por la raíz y se descartan todas aquellas ramas con rótulo *i* tal que $i \notin [d(p, q)-r, d(p, q)+r]$ siendo *p* el pivote utilizado en la raíz. La búsqueda continúa recursivamente en todos aquellos subárboles no descartados, utilizando el mismo criterio.

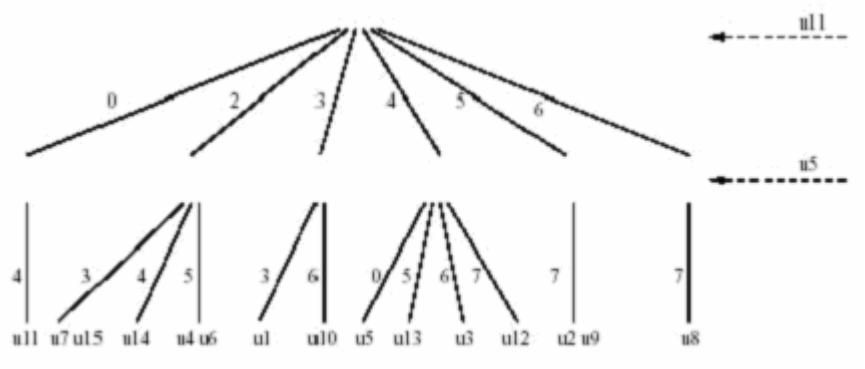


Figura 1: FHQT con pivotes u11 y u5

2.3 Bases de Datos Métrico-Temporales

Las Bases de Datos Métrico-Temporales permiten realizar búsquedas sobre objetos que tienen un intervalo de vigencia, y que no poseen un identificador que se pueda utilizar como clave de búsqueda, por lo cual tiene sentido consultar por similitud y considerando también el aspecto temporal.

Formalmente se define como sigue [15]: sea *O* el universo de objetos válidos, un **Espacio Métrico-Temporal** es un par *(U, d)*, donde el conjunto $U=O \times N \times N$ y la función *d*, es de la forma $d: O \times O \rightarrow R^+$. Cada elemento $u \in U$ es una 3-upla (obj, t_i, t_f), donde *obj* es un objeto (por ejemplo, una imagen, sonido, cadena, etc) y $[t_i, t_f]$ es el intervalo de vigencia de *obj*. La función de distancia *d*, que mide la similitud entre dos objetos, cumple con las propiedades de una métrica (positividad, simetría, reflexividad y desigualdad triangular).
Sea $X \subseteq U$ el conjunto finito sobre el que se realizan las búsquedas, una consulta métrico temporal se denota mediante la 4-upla $(q, r, t_{iq}, t_{fq})_d$, y se define formalmente de la siguiente manera:

$$(q, r, t_{iq}, t_{fq})_d = \{ o \ / (o, t_{io}, t_{fo}) \in X \ \wedge \ d(q, o) \leq r \ \wedge (t_{io} \leq t_{fq}) \wedge (t_{iq} \leq t_{fo}) \}$$

La variable *r* es el radio de búsqueda y representa el grado de similitud con que se consulta, y *q* es el objeto de consulta. Respecto al tiempo, esta clase de consulta tiene éxito sólo para los objetos cuyo intervalo de vigencia se superpone en algún punto con el intervalo consultado. En el caso de una consulta instantánea (time-slice), $t_{iq} = t_{fq}$.

3 Método de Acceso Métrico-Temporal: el Historical FHQT

En este artículo se propone una nueva estructura de acceso métrico-temporal denominada Historical FHQT (HFHQT), que utiliza el índice métrico FHQT [1] para considerar el aspecto métrico, e ideas de índices temporales para tratar el aspecto temporal.

El HFHQT consiste en una lista de los instantes válidos de tiempo, donde cada celda contiene un índice FHQT de todos los objetos vigentes en dicho instante. Esta estructura, que puede considerarse trivial, es eficiente en bases de datos métrico-temporales en donde los objetos tienen vigencia en un sólo instante de tiempo.

3.1 Estructura

Formalmente, un HFHQT es un par (*li*,*lp*) donde:

- *li* es una lista (f_1, f_2, ..., f_n) en la cuál [1, *n*] es el intervalo válido de tiempo, y cada f_i es un FHQT correspondiente al instante *i*, o *nil* si no hay ningún objeto vigente en dicho momento.
- *lp*=(p_1, p_2, …, p_{max}) es la lista de pivotes utilizados en la construcción de todos los árboles. *max* es la cantidad de pivotes del árbol más profundo de la lista.

Los árboles pueden tener distintas profundidades de acuerdo a la cantidad de elementos que se deban indexar. La cantidad de pivotes utilizada en un árbol se calcula como $\lceil \log_2(|o_i|) \rceil$, donde $|o_i|$ es la cantidad de objetos vigentes en el instante *i*. De esta manera se evita que haya árboles profundos cuando la cantidad de objetos es baja, con el fin de que la estructura no tenga un costo espacial excesivo.

El valor *max* se utiliza durante la consulta para determinar el tamaño de la firma del objeto consultado. La lista *lp* contiene los pivotes que se utilizan en todos los árboles; el pivote p_i es el pivote correspondiente al nivel *i* de los árboles que poseen al menos dicho nivel.

La estructura es dinámica, permitiendo altas de dos tipos: históricas o de nuevos instantes.

- Un alta es histórica cuando se incorpora un objeto a un instante ya existente. El costo de este tipo de alta es el costo de calcular la firma del nuevo objeto, siempre que no haya que reestructurar el árbol.
- Un alta de un nuevo instante, toma como entrada un conjunto de objetos, construye el FHQT correspondiente a dicho conjunto, y lo agrega al final de la lista *li* como nuevo instante. Los instantes en los cuales no haya objetos vigentes, deben ser agregados a *li* a través de conjuntos vacíos, y se representan dentro de la lista con el valor *nil*.

3.2 Consulta

Las consultas métrico temporales se efectúan de la siguiente manera: en primer lugar se seleccionan de la lista *li* los instantes incluidos en el intervalo de consulta. Posteriormente se realizan consultas por similitud usando cada uno de los FHQT correspondientes, y se unen los conjuntos resultantes.

En la Figura 2 se muestra un ejemplo del HFHQT. El intervalo total representado es [1, 12]. La estructura contiene FHQTs sólo en los instantes 2, 5, 7, 8 y 9, ya que en el resto no hay objetos

vigentes registrados. Como se ve, los árboles no tienen todos la misma profundidad: los correspondientes a los instantes 2 y 9 utilizan sólo un pivote; los correspondientes a los instantes 7 y 8 usan dos pivotes y el 5 contiene un FHQT de 3 pivotes. En el ejemplo $lp = (p_1, p_2, p_3)$ y max=3.

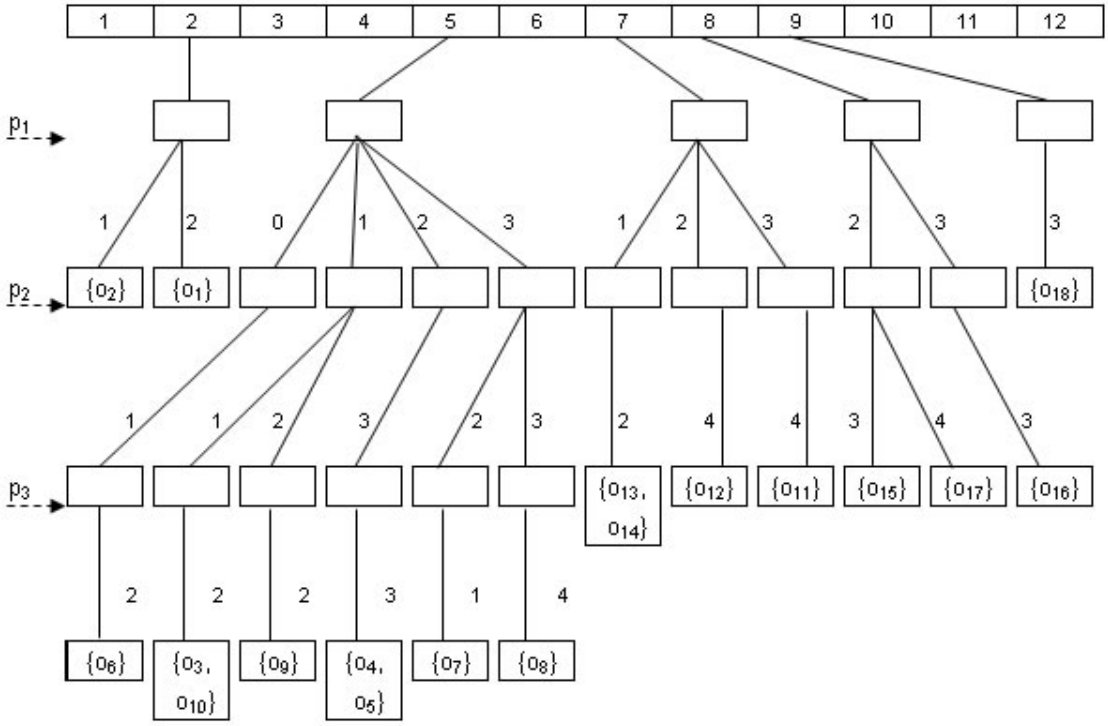


Figura 2: Historical FHQT

Para la consulta $(q, 1, 6, 9)_d$, siendo *(1, 2, 4)* la firma de *q*, el HFQT se comporta de la siguiente manera: se acceden en forma directa los instantes 6, 7, 8 y 9 y se busca por similitud en los FHQT correspodientes. El instante 6 se ignora ya que no posee un FHQT asociado. En el instante 7, como la firma de *q* contiene como primer elemento a 1, y el radio de la búsqueda es 1, se toman las ramas 1 *(1-1≤1≤1+1)* y 2 (*1-1≤2≤1+1*), y se descarta la 3. En el siguiente nivel, sólo cumple la restricción la rama 2, por lo cual los objetos candidatos son $\{o_{13}, o_{14}\}$. El mismo proceso se ejecuta para los instantes 8 y 9, dando como candidato a $\{o_{15}\}$. Luego estos conjuntos se unen en $\{o_{13}, o_{14}, o_{15}\}$ y los objetos contenidos se comparan con la consulta *q* para obtener el resultado final.

3.3 Algoritmo de consulta

En la Figura 3, se muestra el pseudocódigo del algoritmo de consulta al HFHQT. La función toma como entrada el objeto de consulta *q*, el radio de búsqueda *r*, y el intervalo que se consulta $[t_{iq}, t_{fq}]$. Como primer paso se calcula la firma de *q* para todos los pivotes utilizados en al menos un árbol del HFHQT; luego se consulta cada FHQT correspondiente al intervalo consultado y se unen los conjuntos resultantes para obtener el resultado final

HFHQT$(q, r, t_{iq}, t_{fq})_d$
begin
  calcular la firma f de q -- *distancias a cada uno de los pivotes de lp*
  candidatos:={}
  for all (fhqt h correspondiente a los instantes del intervalo $[t_{iq}, t_{fq}]$ del HFHQT)
    candidatos :=candidatos $\cup h(q, r, f)_d$
  resultado:={$x \in$ candidatos | $d(q, x) \leq r$ }
  return resultado
end.

Observación: $h(q, r, f)_d$ devuelve el conjunto de objetos candidatos a ser similares a q con radio r a través de una consulta al FHQT h, donde f es la firma de q.

Figura 3: Pseudocódigo de consulta del HFHQT

4 Resultados Experimentales

Para verificar su funcionamiento se realizó la implementación del HFHQT y se probó sobre una base de datos de 750 imágenes representadas a través de vectores de *762* dimensiones [9] a las cuales se les agregó un número natural que representa el instante de validez de la imagen. El intervalo total considerado fue *[1, 100]*. Se utilizó la distancia coseno discretizada [16, 8] como función de distancia, y los costos se expresaron en cantidad de evaluaciones de dicha función. No se evaluaron otras variables que podrían tener influencias importantes en los resultados, como por ejemplo, la cantidad de accesos a disco.

Los árboles correspondientes a cada instante de tiempo se construyeron tomando de una lista de pivotes elegidos al azar [6]. La cantidad de pivotes de cada árbol se calculó como $\lceil \log_2(|O_i|) \rceil$, donde $|O_i|$ es la cantidad de objetos vivos en el instante i. En cada prueba se ejecutaron y promediaron *100* consultas variando el radio de búsqueda (*1, 3 y 10*), el tamaño promedio del intervalo de consulta (instantánea; 10%, 25% y 50% del intervalo total), y la cantidad de elementos de la base de datos (*100, 250, 500 y 750*).

La comparación se realizó contra la solución trivial planteada en [15], que consiste en construir un FHQT para toda la base de datos y ante una consulta, realizar la búsqueda por similitud utilizando dicho índice, para luego eliminar del conjunto resultante los elementos que no cumplen con la restricción temporal. En esta solución, el costo está dado sólo por la cantidad de evaluaciones de la función de distancia en las consultas al FHQT, y es el mismo tanto para consultas instantáneas como para consultas por intervalos.

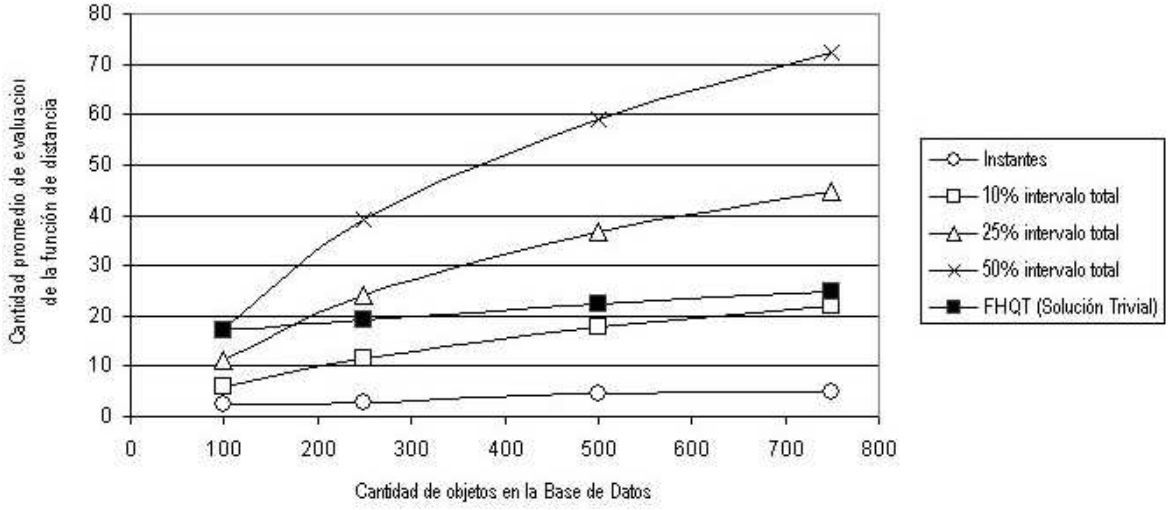


Figura 4: Historical FHQT, radio 1

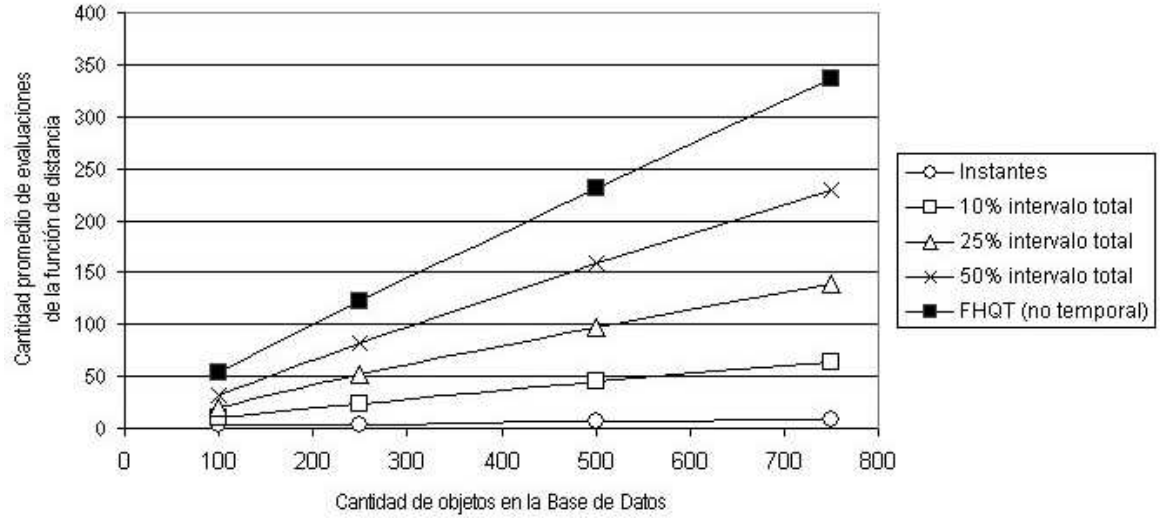


Figura 5: Historical FHQT, radio 3

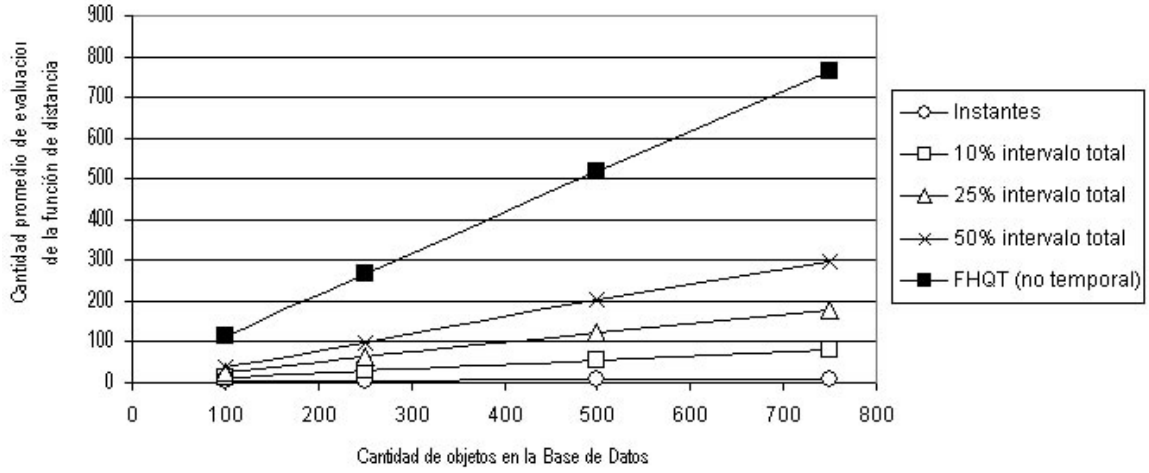


Figura 6: Historical FHQT, radio 10

En las Figuras 4, 5 y 6 se muestran las cantidades promedio de evaluaciones de la función de distancia ante consultas al HFHQT y de la solución trivial, para los radios de búsqueda *1, 3* y *10* respectivamente.

Cuando la consulta es instantánea (time-slice) el HFHQT obtiene sus mejores resultados, alcanzando entre un 524% y un 7628% de disminución del costo sobre la solución trivial. Esto es debido a que mientras en la solución trivial se debe consultar un árbol que contiene todos los objetos de la base de datos, en el HFHQT sólo es necesario acceder al árbol que contiene los objetos vigentes en el instante requerido en la consulta, que normalmente es mucho mas chico.

Como se ve, en las consultas por intervalo el HFHQT también se comporta mejor que la solución trivial, excepto para radio 1 con intervalos del 25 o 50% del tiempo total. En estos casos, la solución trivial es más eficiente ya que la capacidad de filtrado de un único FHQT que contiene todos los objetos de la base de datos y por lo tanto, que posee mayor cantidad de niveles (pivotes), es superior a la de varios FHQTs de menor profundidad. Se verificó experimentalmente que la eficiencia del HFHQT en estos casos puede ser mayor a la de la solución trivial, si se utiliza una cantidad fija, igual a la máxima, de pivotes para todos los árboles.

Costos

Un HFHQT es muy eficiente ante consultas por similitud instantáneas o por intervalos reducidos. Sea *dt* la densidad temporal (cantidad promedio de objetos por instante de tiempo), el costo de una consulta $(q, r, t_{iq}, t_{fq})_d$ al HFHQT es $c(t_{fq} - t_{iq}+1)$ donde *c* es el costo de la consulta $(q, r)_d$ a un FHQT con *dt* cantidad de objetos y $\lceil \log_2(|dt|) \rceil$ niveles [1, 2, 3]. En una consulta por similitud instantánea, *c* se reduce a 1, es decir que se consulta sólo un árbol, y de tamaño reducido en comparación con el FHQT de la solución trivial planteada. Si *n* es la cantidad de objetos de la base de datos, en este caso se descartan (*n-dt*) objetos en forma directa, sin realizar ningún cálculo de la función de distancia.

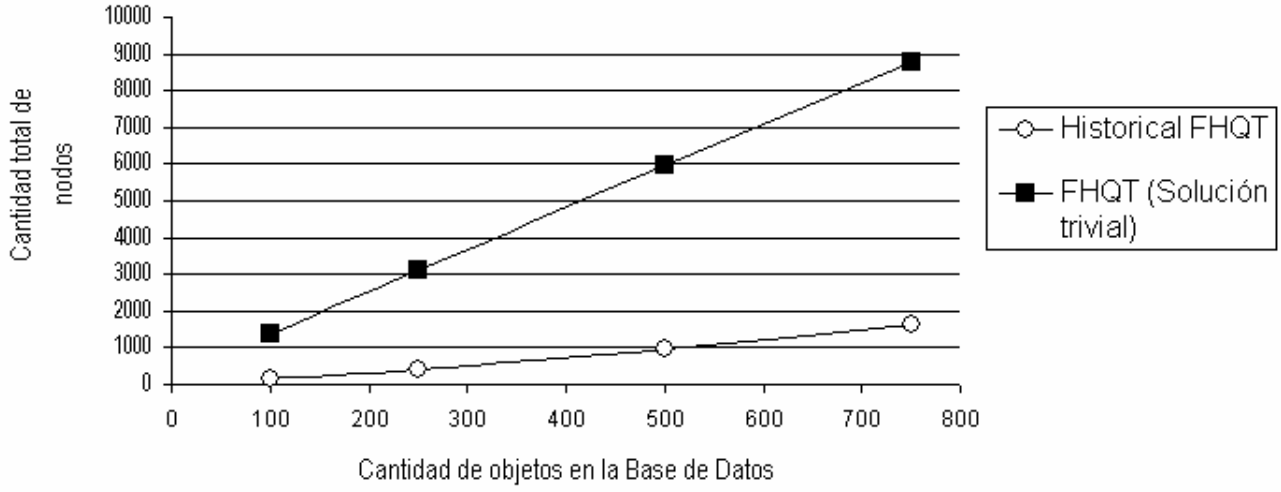


Figura 7: Costo Espacial del Historical FHQT

El HFHQT tiene menor costo espacial que el FHQT de la solución trivial, si se sigue la estrategia de utilizar cantidades logarítmicas de pivotes para los árboles correspondientes a cada instante. Los resultados se presentan en la Figura 7, donde se compara la cantidad total de nodos de los árboles del HFHQT, con la cantidad de nodos de un único FHQT para 100, 250, 500 y 750 objetos. Esto se debe a que varios árboles pequeños ocupan menos espacio que un solo árbol de mayor profundidad,

ya que normalmente el incremento en nodos de un nivel al siguiente es exponencial. En caso de utilizar una cantidad fija de pivotes para todos los árboles, igual a la de la solución trivial, el costo espacial del HFHQT alcanza a ser hasta el doble que el de un único FHQT.

5 Conclusiones y Trabajo Futuro

En este trabajo presentamos un nueva estructura -el Historical FHQT-, orientada a resolver consultas métrico-temporales instantáneas sobre una base de datos de objetos instantáneos. El método se comporta eficientemente en estos casos, aunque también puede ser utilizado para consultar intervalos de tiempo, pero con menor eficiencia.

Los resultados de los experimentos realizados muestran que los costos de consultas por similitud instantáneas o por intervalos pequeños son significativamente menores que los de la solución trivial planteada. Esta eficiencia se logra debido a que se consulta una cantidad reducida de árboles de pocos elementos porque se descartan en forma directa todos los objetos que no estuvieron vigentes en el instante o intervalo requerido.

Una desventaja de este método es que las consultas métricas puras tienen mayor costo que el mismo tipo de consultas a un FHQT. Por otro lado, es de notar que en una consulta temporal pura el acceso a los instantes es directo, y los objetos se recuperan a través de un recorrido a los árboles correspondientes.

Este nuevo índice se presenta como un primer avance de una solución más completa en la que estamos trabajando actualmente. En la misma, introducimos modificaciones al HFHQT tomando ideas del HR-Tree, para mejorar su eficiencia ante consultas por intervalos, y permitir representar objetos que tengan asociado un intervalo de vigencia en lugar de instantes.

También estamos organizando nuevas pruebas sobre una base de datos de 30000 imágenes con el fin de verificar el comportamiento del índice en situaciones más cercanas a la realidad. Sobre esta base haremos pruebas de eficiencia tomando en cuenta la variable “cantidad de accesos a disco” como un factor más en el establecimiento de los costos, ya que consideramos que puede tener una influencia importante en los resultados.

Referencias

[1] Baeza-Yates, R., Cunto, W., Manber, U. and Wu S. Proximity matching using fixed-queries trees. In *Proc. 5th Combinatorial Pattern Matching (CPM’94),* LNCS 807, pages 198–212, 1994.

[2] Baeza-Yates, R. Searching: an algorithmic tour. In A. Kent and J. Williams, editors, *Encyclopedia of Computer Science and Technology*, volume 37, pages 331–359. Marcel Dekker Inc., 1997.

[3] Baeza-Yates, R., Navarro G. Fast Approximate String Matching in a Dictionary. In Proceedings of SPIRE'98, pages 14-22. String Processing and Information Retrieval: a South American Symposium, IEEE Computer Society, Sept. 1998

[4] Bozkaya, T., Ozsoyoglu, M. Distance-based indexing for high-dimensional metric spaces. In *Proc. ACM SIGMOD International Conference on Management of Data*, pp. 357–368, Sigmod Record 26(2), 1997.

[5] Brin, S. Near neighbor search in large metric spaces. In *Proc. 21st Conference on Very Large Databases (VLDB'95)*, pages 574–584, 1995.

[6] Bustos, B., Navarro, G. and Chávez, E. Pivot selection techniques for proximity searching in metric spaces. In *Proc. of the XXI Conference of the Chilean Computer Science Society (SCCC'01)*, pages 33–40. IEEE CS Press, 2001.

[7] Chávez, E., Navarro, G., Baeza-Yates, R. and Marroquín, J.L. Searching in metric spaces. In *ACM Computing Surveys*, 33(3):273.321, September 2001.

[8] Chávez, E., Herrera, N., Ruano, C., Villegas, A. Una implementación completa del FQTrie. *VII Workshop de Investigadores de Ciencias de la Computación*, pp 61-65. 2005.

[9] Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D. and Equitz, W. Efficient and effective querying by image content. *Journal of Intelligent Information Systems (JIIS)*, 3(3/4):231–262, 1994.

[10] Guttman, A. R-trees: A dynamic index structure for spatial searching. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pp. 47-54. 1984

[11] Jensen, C.S. editor et al. A Consensus Glossary of Temporal Database Concepts. In *ACM SIGMOD Record.* 23(1):52-64. 1994.

[12] Kumar, A., Tsotras, V.J., Faloutsos, C. Designing Access Methods for Bitemporal Databases. IEEE Transactions on Knowledge and Data Engineering. 10(1):1-20. 1998.

[13] Navarro, G. A guided tour to approximate string matching. ACM Computing Surveys (CSUR), 33(1):31–88, 2001.

[14] Ozsoyoglu, G., Snodgrass, R.T. Temporal and Real-Time Databases: A Survey. IEEE Trans. on Knowledge and Data Engineering. 7(4):513-532, 1995.

[15] Pascal, A., De Battista, A., Gutiérrez, G., Herrera, N. Procesamiento de Consultas Métrico-Temporales. Artículo aceptado para su presentación en la XXXIII Conferencia Latinoamericana de Informática (CLEI), 2007.

[16] Ruano, C., Chávez, E., Herrera, N. Discretización binaria del FQtrie. In Actas del X Congreso Argentino de Ciencias de la Computación (CACIC'04), pages 100–111, Buenos Aires, Argentina, 2004.

[17] Salzberg, B., Tsotras, V.J. A Comparison of Access Methods for Temporal Data. ACM Computing Surveys. 1999.

Una implementación de un sistema de control domótico basada en servicios web

Mauricio Esteban Pardo **Guillermo Enrique Strack** **Diego C. Martínez**
mepardo10@hotmail.com **guillotest@gmail.com** **dcm@cs.uns.edu.ar**

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur,
Avenida Alem 1253 Bahia Blanca, Buenos Aires, Argentina

Abstract

The aim of this project is to show the development of a Domotic System to control differents electrical devices in a house either from inside the building or by remote control using the Internet.
In order to have these two functions available, the system includes a Server that provides Web Services to the controlling applications in the house. This Server also offers an interface that uses AJAX to optimize device handling. Finally, considerations about how to generate executing turn on/off routines of electrical household appliances using machine learning tools are presented

Key Words: Home Automation, domotic, Web Services , AJAX.

Resumen

El objetivo de este trabajo es presentar una implementación de un Sistema Domótico para controlar diferentes dispositivos eléctricos de una vivienda tanto en forma local como así también en forma remota, utilizando una conexión a Internet hogareña. Para permitir estos dos puntos de acceso, el sistema incluye un servidor que provee servicios web a las aplicaciones controladoras instaladas en el hogar. Este servidor también ofrece una interfaz que utiliza AJAX para optimizar la manipulación de los dispositivos. Finalmente, se presentan consideraciones sobre la generación de rutinas ejecutables de encendido y apagado de los electrodomésticos utilizando herramientas de aprendizaje automatizado.

Palabras claves: Automatización del hogar, domótica, Servicios Web , AJAX.

1. INTRODUCCION

Los sistemas domóticos son sistemas inteligentes para casas y departamentos, que integran y controlan áreas tan diversas como las comunicaciones, la informática, la seguridad, la iluminación, ambientación climática, y los electrodomésticos, generando considerables beneficios en lo que a confort, seguridad y ahorro de energía se refiere [3,4,5]. Actualmente existen varias empresas que se dedican a la implementación de estos sistemas. Dependiendo de cada solución o fabricante, hay productos que son controladores, sensores y actuadores al mismo tiempo, ya que en un único equipo se dispone de toda la inteligencia necesaria para medir una variable física, procesarla y actuar en consecuencia (por ejemplo, un termostato). Sin embargo, la mayoría de las soluciones del mercado, sean propietarias o no, se construyen diferenciando los sensores de los actuadores con el objeto de aportar mayor flexibilidad y menor precio, de cara a la instalación e integración en una vivienda.

Los *servicios Web o Web Services* [1,6] son una metodología que permite intercomunicar dos sistemas remotos a través de la Web, manteniendo el estado de los objetos en la transferencia. En términos generales, un Web Service nos permite ejecutar un método remotamente y recibir su resultado como si se tratara de un método local a nuestro sistema. Ellos se basan en ciertos protocolos estándar, como SOAP y XML, procurando formalizar la intercomunicación entre sistemas. Gracias a esto, es posible comunicar, por ejemplo, un sistema de ASP.NET ejecutando sobre IIS con uno desarrollado en Java en un Servidor Solaris.

AJAX es un acrónimo de *Asynchronous JavaScript And XML* [9], una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan del lado cliente, es decir, en el navegador del usuario, y mantiene una comunicación asincrónica con el servidor en segundo plano. De esta forma es posible, por ejemplo, realizar cambios visuales sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y la usabilidad de la interfaz web.

En este trabajo se presenta la implementación de un sistema domótico Controlador-Actuador, desarrollado utilizando un Servicio Web para lograr el acceso remoto del mismo y AJAX en la interfaz del mismo. Con el fin de lograr simplicidad en su instalación, el control de los dispositivos a manipular es llevado a cabo mediante comunicación inalámbrica por radiofrecuencia como así también comunicación cableada.

El trabajo se organiza de la siguiente manera. En la Sección 1 se presenta el escenario adoptado y las premisas para el desarrollo. En la Sección 2 se muestran los elementos principales de la arquitectura del sistema y la interacción entre ellos. En las secciones subsiguientes se revisan los trabajos relacionados y se presentan extensiones al sistema.

2. ESCENARIO DE DESARROLLO

De acuerdo a las necesidades específicas, un sistema domótico puede ser implementado de diversas formas y, como es natural, en base a las funcionalidades y capacidades que el mismo brinde, se determina su costo. En nuestro caso, se hizo mayor hincapié en este último factor y por ende el sistema implementado provee una funcionalidad puntual: permite controlar y programar remotamente los diferentes dispositivos electrónicos que puedan existir en una vivienda o comercio común. Varias premisas son importantes para la arquitectura propuesta:

- El sistema debe permitir que los dispositivos, además de poder ser controlados localmente, puedan ser manipulados remotamente, contribuyendo a la automatización del hogar [7].
- El sistema debe permitir la administración centralizada de múltiples viviendas independientes entre sí, probablemente de usuarios diferentes.
- El sistema debe utilizar tecnologías fácilmente asimilables por el público, sin requerir gran poder adquisitivo, para facilitar su acceso e implementación.

En función de estas premisas iniciales, la arquitectura propuesta requiere un servidor Web que centraliza las viviendas a controlar y provee la funcionalidad necesaria para administrar los

dispositivos remotamente. Se requiere, además, una computadora en la vivienda que se conecta con los dispositivos controlables (lámparas, televisor, etc). Para simplificar la implementación de la aplicación hogareña, ésta no requiere un número de IP fijo, por lo que la aplicación hogareña actuará exclusivamente como cliente del servidor Web. De esta manera el usuario no requiere más que un común acceso a Internet. En la Figura 1 se representa esquemáticamente la arquitectura simplificada.

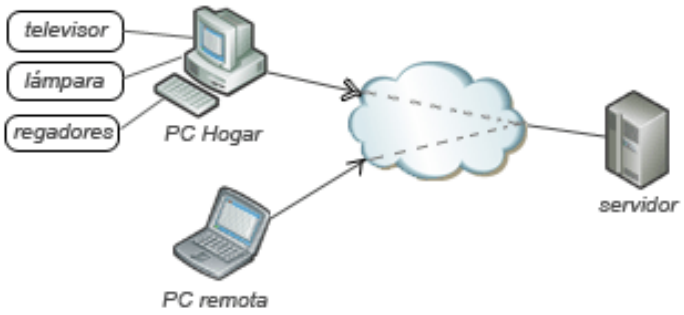


Figura 1. Componentes principales de la arquitectura

La computadora instalada en la vivienda (*PC Hogar* en la Figura 1) permite controlar, por medio de una interfaz adecuada, los dispositivos conectados a ella. Es el controlador *in-situ* de una vivienda individual. El servidor externo, a su vez, posee la misma funcionalidad con el fin de poder ser accedido remotamente. Es el controlador *remoto* de varias viviendas administradas por usuarios.

Esta implementación obliga a disponer de dos bases de datos, una general con la información de todos los usuarios que utilicen este servicio, ubicada en el servidor Web, y otra base de datos local, almacenada en la pc local del usuario con la información referida a su casa solamente.

Esta doble funcionalidad en la vivienda y en el servidor es deseable desde el punto de vista de la accesibilidad, pero requiere un esfuerzo adicional ante ciertos requerimientos básicos, como mantener la consistencia entre la base de datos ubicada en el servidor y la base de datos alojada localmente. Para ello es necesario desarrollar un protocolo de sincronización que permita mantener las dos bases de datos actualizadas antes los cambios que ocurran desde cualquiera de los dos lados.

Para implementar ese protocolo se utiliza un *web service*, que permite que la casa pueda indagar la base de datos del servidor. Es importante destacar también un beneficio de esta duplicación: más allá del problema de mantener la consistencia de la base de datos, el sistema otorga la comodidad de manejar los dispositivos de la casa sin contar necesariamente con una conexión a Internet. En este caso, las modificaciones se hacen localmente y luego, cuando se disponga de la conexión, se actualizan las dos bases de datos acordemente.

El hardware utilizado para comunicar la aplicación de la computadora *in-situ* con los dispositivos se conecta por el puerto paralelo. Las órdenes del puerto hacia el destinatario final (los elementos eléctricos a controlar) pueden ser enviadas tanto en forma *cableada* como así también en forma *inalámbrica por radiofrecuencia*, facilitando así la tarea de instalación del sistema en una vivienda. Este aparato que interconecta la computadora del hogar con los dispositivos fué desarrollado y construido por los autores y su estructura es explicada más adelante.

En la siguiente sección se describe más detalladamente las funcionalidades que disponen el software, el modo de operación y el hardware implementado en esta ocasión.

3. COMPONENTES DE LA APLICACION

El sistema desarrollado está constituido por diversos participantes. Entre ellos se encuentran el *Software de Diseño de Interfaz de la Vivienda*, el *Software de Control de los Dispositivos* (Aplicación Local) y la *Aplicación Web para Control de los Dispositivos* (Aplicación Remota) .

3.1 Software de Diseño de Interfaz de la Vivienda

Esta aplicación provee las herramientas necesarias para crear una interfaz gráfica adecuada para representar la vivienda de manera personalizada. Aquí el usuario podrá seleccionar los ambientes que componen su casa. Cada uno de ellos va a estar representado por un nombre y una imagen significativa del mismo, y a su vez ir ubicando en ellos los dispositivos que componen un ambiente y se deseen manipular. En vistas de la extendibilidad, esta aplicación le da al cliente la posibilidad de ingresar posteriormente nuevos ambientes y dispositivos, como así también la capacidad de editarlos o eliminarlos. En esta etapa el dueño de la casa definirá un nombre de usuario y una contraseña los cuales serán empleados para iniciar sesiones futuras antes posibles cambios en el diseño de la casa o en sesiones de la aplicación local o remota. Una vez que el usuario termine de diseñar su vivienda, los cambios se verán reflejados tanto en la base de datos del servidor como en la base de datos local. En la Figura 2 se muestra una captura de pantalla de una casa ya configurada. La disposición de las habitaciones y los elementos incluidos en ellas (lámparas, televisores, radios, etc) son completamente configurables. La aplicación también permite configurar dispositivos exteriores, como las luces de entrada o los regadores del jardín. El objetivo esencial de esta aplicación es la facilidad de uso en vistas a usuarios no experimentados.

3.2 Software de Control de Dispositivos

La función de ésta aplicación es proveer acceso a los diferentes componentes modelados visualmente en la fase de *Diseño de Interfaz*. Aquí el usuario podrá alterar el estado real de los dispositivos, como apagar o prender una luz, el televisor, activar los sensores perimetrales o el alimentador de mascotas. Puede también ingresar rutinas pre-programadas especificando los dispositivos involucrados, el estado que van a tener al momento de ejecutarse la rutina y el periodo de ejecución de la misma (una vez al día, una vez a la semana, etc). Estas rutinas permiten simular movimiento si el usuario está fuera de la casa o planificar tareas como el riego del jardín. Una vez más, cualquier cambio reflejado aquí, (cambio de estado de un dispositivo o ingreso, modificación de alguna rutina), se verá reflejado tanto en la base de datos local como también en la base de datos remota, la que está ubicada en el servido Web.

Figura 2. Interfaz para el control de los dispositivos

3.3 Aplicación Web para Control de los Dispositivos

Mediante la aplicación web los usuarios del sistema podrán iniciar una sesión y realizar las mismas tareas que puede ejecutar con la aplicación anterior. Esto es: *encendido* y *apagado* de los distintos dispositivos como así también *ingreso*, *modificación* y *eliminación* de las rutinas preprogramadas. Cabe destacar que la sesión en la Web va a poder ser iniciada únicamente cuando el programa anterior (Control De Dispositivos) esté conectado al servidor, es decir que la computadora *in-situ* disponga conexión a Internet activa y la aplicación se esté ejecutando. Esta restricción es naturalmente necesaria para la interacción entre el hogar y el servidor. De lo contrario, podría ocurrir que remotamente se intente cambiar el estado de un dispositivo hogareño cuando es imposible que la orden llegue a los mismos. Un aspecto importante a remarcar es que las acciones indicadas remotamente por medio de esta aplicación web no son comunicadas a la computadora del hogar, sino que es ésta quien indaga al servidor sobre los cambios requeridos. Esta comunicación unidireccional es explicada en la siguiente sección

4. FUNCIONAMIENTO DEL SISTEMA

El funcionamiento del sistema se basa en dos tipos de comunicaciones para lograr consistencia de las base de datos y así reflejar los eventos realizados en los distintos participantes del sistema (Aplicación local y Aplicación Web). Deben considerarse los dos extremos como iniciadores de eventos de modificación: una de estas comunicaciones se da cuando desde la aplicación local se altera el estado de los dispositivos y la otra cuando por medio del sitio web se demandan esos cambios.

En el caso de que los cambios se produzcan desde la aplicación local, ya sea sobre los dispositivos o sobre alguna rutina, se recolecta información de este suceso y se envía hacia el servidor en forma inmediata. Para ello se utiliza un servicio web [6], el cual realiza la actualización sobre la base de datos del servidor. Posteriormente el cambio se refleja en la base de datos local y a continuación el dispositivo o rutina especificada cambian su estado. En la Figura 3 se muestran los pasos generales de este proceso. En los casos que el usuario no disponga de conexión a Internet o no le interese por el momento que las modificaciones también sean reflejadas en el servidor, los cambios son solo aplicados en la base de datos local y luego reflejados en los dispositivos reales (apagar una luz, encender el sistema de riego, etc). Esto genera una pequeña inconsistencia momentánea con la base de datos del servidor, pero la sincronización postergada se inicia inmediatamente cuando la aplicación vuelve a poseer conexión con el servidor de la aplicación web.

De esta manera el uso local del sistema no se ve perjudicado ante cualquier problema que ocurra con el proveedor de Internet (a veces frecuentes) u algún otro motivo que inhabilite la comunicación entre las dos partes.

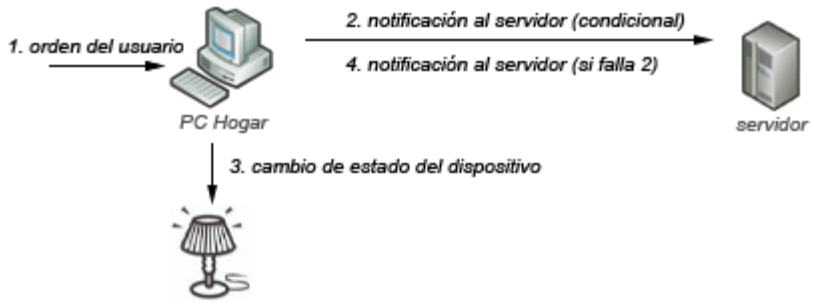


Figura 3. Cambios desde la computadora del hogar.

Para el caso de que los cambios se realicen desde la página web, hay leves diferencias, principalmente porque el sitio web no se comunica con la aplicación local que está corriendo en la computadora de la casa del usuario. De hecho, la aplicación local es la que se comunica con la aplicación web y esta particularidad es la que permite que el usuario no necesariamente tenga un número de IP público para su salida a Internet, pudiendo el sistema ser utilizado desde una red privada. El acceso unidireccional es además seguro y simple para un hogar promedio. En la Figura 4 se muestran los pasos generales de este proceso. Al momento en que el usuario decide modificar remotamente (vía la aplicación web) el estado de algún dispositivo o modificar la configuración de alguna rutina, éste cambio se registra únicamente en el servidor. La aplicación local realiza periódicamente una consulta al servidor, el cual implementa un servicio web indicando si hubo cambios registrados en su propia base de datos.

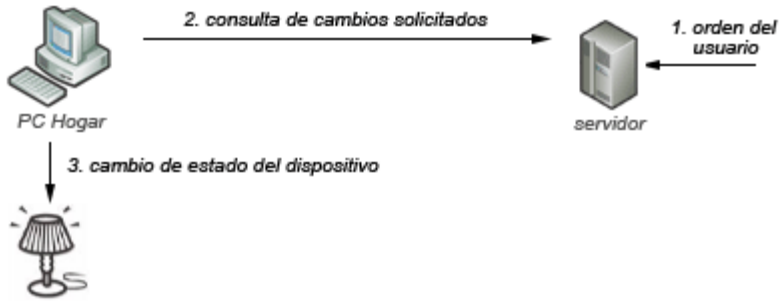


Figura 4. Cambios desde el servidor web

Esta consulta periódica por parte de la aplicación se resuelve por intermedio de dos hilos de ejecución, denominados `HiloControlDispositivos` e `HiloControlRutinas`, respectivamente. El primero controla si hubo cambios en los estados de los dispositivos, y el segundo controla acciones relacionadas a la manipulación de las rutinas. En el caso puntual del `HiloControlDispositivos`, éste solamente solicita un servicio web que devuelve un vector con los identificadores de los dispositivos que están activos en la base de datos del servidor. Luego se activan los dispositivos que están en este vector y se deshabilita el resto, se hacen las modificaciones correspondientes sobre la base de datos local y luego sobre los dispositivos reales a través del puerto paralelo. Para determinar si se realizaron cambios en las rutinas, el `HiloControlRutinas` solicita un servicio web que le retorna un vector con las rutinas activas (formado por los identificadores de rutinas activas), y un indicador para controlar si se han ingresado remotamente nuevas rutinas y/o si se modificaron parámetros de rutinas existentes. En este último caso, si se hubieren ingresado o modificado rutinas, el hilo ejecuta un nuevo servicio Web solicitando información correspondiente a las nuevas rutinas ingresadas o modificadas según corresponda. Estas modificaciones o rutinas nuevas son cargadas en la base de datos local y luego se procede a activar aquellas rutinas indicadas por el servicio web y desactivar aquellas que no están en el vector devuelto por éste. En la Figura 5 se muestra esta interacción entre el cliente y el servidor.

Los procedimientos descriptos anteriormente se repiten periódicamente. El parámetro que indica el período de actualización puede ser configurado por el usuario. Básicamente, un valor bajo permite visualizar los cambios realizados desde la página en forma mas rápida pero con la desventaja de que la cantidad de requerimientos a los servicios web se incrementa. Un valor alto, en cambio, disminuye la carga en las comunicaciones pero prolonga el período de potencial inconsistencia en las bases de datos.

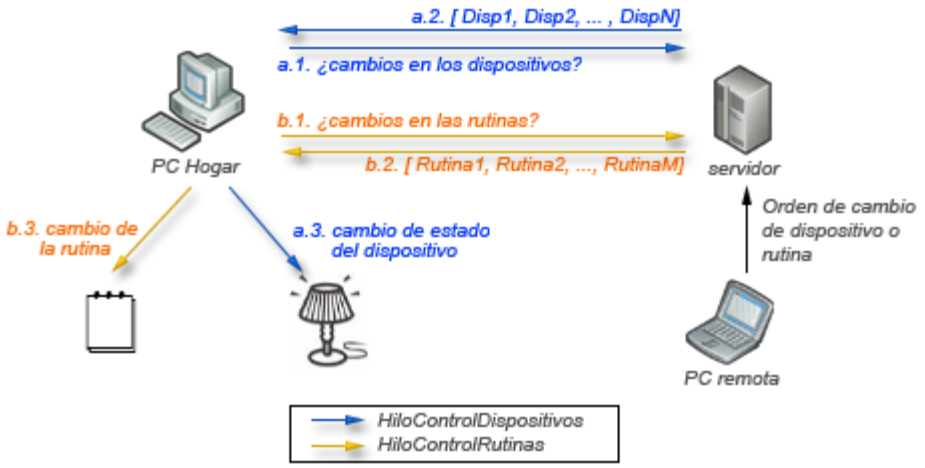


Figura 5. Hilos de Control de Dispositivos y Rutinas

4.1 Interfaz visual de comandos de dispositivos del hogar

La utilización del sistema desde la Web es intuitivo al hacer uso de la técnica AJAX [9], que permitió brindar una apariencia similar al uso de la aplicación ubicada en la casa. Teniendo en cuenta la interfaz utilizada donde varias imágenes muestran los ambientes y los dispositivos instalados en la casa (como indicado en la Figura 2), el refresco parcial de la página mejora notablemente la utilización del sistema. Debido a que un dispositivo encendido tiene una imagen diferente a un dispositivo apagado, al momento de realizar una acción sobre el dispositivo, la actualización de sólo la pequeña imagen asociada al dispositivo seleccionado (mostrar la imagen de encendido o apagado) brinda una mayor performance cuando usamos el sistema desde la web. El mismo efecto se pudo lograr para el caso de ingresar o editar rutinas preprogramadas, toda la carga de las opciones de selección, que se realizan obteniendo datos desde la base de datos, al utilizar AJAX se mejora la usabilidad, ya que a medida que se van seleccionando las opciones solo se van refrescando los controles correspondientes y no toda la página como ocurriría si se hubiese respetado una implementación clásica. En la Figura 5 puede verse esquemáticamente la interacción principal en este escenario. Existen dos niveles de comunicación del cliente al servidor, marcados con flechas de diferente color. La flecha verde denota un envío de comandos de manipulación de dispositivos. Esto sucede cuando el usuario realiza algún click sobre las áreas de representación de dispositivos (marcadas con rectángulos rojos en la interfaz). Las flechas de color rojo indican una interacción AJAX periódica que examina el estado de la casa para reflejarlo visualmente en la interfaz. Esta interacción está actualmente en desarrollo y evita la necesidad de refrescar la página completa. Al momento de escribir este trabajo los cambios en la interfaz son realizados con anterioridad a la comunicación con el servidor. Es más adecuado, sin embargo, mantener un intercambio asincrónico de datos con el servidor para reflejar el estado de la vivienda y sus dispositivos.

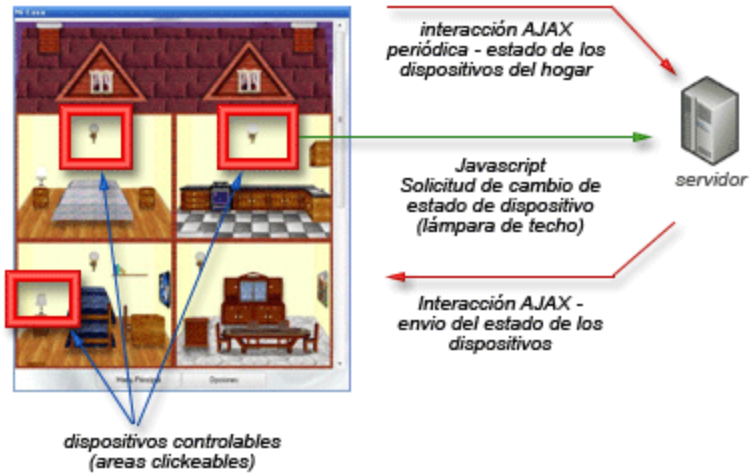


Figura 5. Interfaz visual y AJAX

4.2 Implementación y testeo de la aplicación

La implementación del sistema se realizó íntegramente en la tecnología .NET, en vistas de obtener un prototipo rápidamente. Tanto la Aplicación de Control de Dispositivos como la Aplicación de Diseño de Interfaz para construir la vivienda fueron realizadas con *WinForms* utilizando C#. Para la aplicación Web se utilizó ASP y el *Code Behind* en C#, lo mismo para los Web Services. En cuanto al servidor de base de datos se utilizó *SQLServerExpress.* Los requisitos para la ejecución de este sistema son sólo los requerimientos que tiene el Framework .NET 2.0 y SQLServerExpress. En el aspecto de seguridad, la contraseña del usuario se mantiene encriptada tanto en la base de datos local como así también en la remota, mediante el algoritmo de encriptación provisto por el servidor de base de datos.

En cuanto a la implementación del Hardware se implementaron dos modalidades consideradas útiles y necesarias en un hogar promedio: una vía de comunicación cableada y otra inalámbrica. Esta última brinda una mayor facilidad de instalación para ciertos dispositivos que no son fácilmente trasladables en una habitación. Esta implementación utiliza el módulo transmisor *TWS-434* y el módulo receptor *RWS-434*. Ambos son muy simples de utilizar, su tamaño es muy reducido y con ellos se pueden realizar controles remotos de Radio Frecuencia (RF) a 433.92 Mhz dentro del espectro electromagnético. La modulación en la que se basan es del tipo ASK y pueden ser usados también en otras aplicaciones.

El principio de funcionamiento del hardware desarrollado para la arquitectura propuesta es muy simple. Al intentar cambiar el estado de uno o más dispositivos, se envía al puerto paralelo el código correspondiente a la combinación de dispositivos que se pretende mantener activados y desactivados. Por ejemplo,

`00000001` significa "*activado sólo el dispositivo vinculado a la salida 0 del puerto paralelo*".
`00000101` significa "*activado los dispositivos asociados a la salida 0 y 2 del puerto paralelo*".

Luego la salida del puerto alimentará la entrada de un codificador, el cual generará a su vez la entrada del emisor *TWS-434* para su posterior transmisión. Una vez transmitida, la señal será captada por el receptor *RWS-434*, el cual enviará a un decodificador la señal captada para que este se encargue de convertirla en un código similar al emitido por el puerto paralelo. A ese codificador estarán conectados los relés que están asociados a los dispositivos, que actuarán como un

interruptor permitiendo el paso de corriente eléctrica, activando o desactivando los dispositivos correspondientes. La opción *cableada* no pasa por el emisor. Simplemente, con los datos tomados desde el puerto paralelo, activa los relés correspondientes y estos dejan pasar o interrumpen la corriente para el dispositivo en cuestión, al cual se llega específicamente con un cable que va desde la computadora del hogar.

Se evaluó el correcto funcionamiento de la aplicación en su totalidad, esto es, aplicación Web y aplicación de control. Se puso especial énfasis en la conservación de la consistencia de la base de datos local con la ubicada en el servidor, como así también el hecho de que los métodos brindados por los Web Services tengan un tiempo de respuesta corto. Esta prueba se realizó en horas de mucha carga en la red y el resultado fue satisfactorio. Las base de datos se mantenían inconsistentes por muy poco tiempo (de acuerdo a la configuración de intervalos apropiados mencionados en la Sección 4) y el tráfico generado por los constantes pedidos por parte de la aplicación no influyó en la performance de la red. Se testeó también que las rutinas preprogramadas se ejecuten adecuadamente y al finalizar su ejecución se actualicen para la próxima vez que deban actuar, tanto las realizadas en la propia casa como las creadas desde la página Web.

Otro punto interesante que se evaluó es la frecuencia con la que se solicitan los métodos brindados por el Web Service. Se dejó establecido un parámetro para regular dicha frecuencia. Estas solicitudes son las encargadas de indicarle a la aplicación de control que hay cambios en la casa desde el servidor; en caso de tener una conexión de baja velocidad es apropiado disminuir la frecuencia aplicándole un valor alto al tiempo entre cada solicitud de cambios. En pruebas donde quedaba poco ancho de banda disponible, un valor entre 25 y 30 segundos es admisible. En aquellos casos en los que se pueda hacer un mayor uso del ancho de banda es mejor establecer frecuencias de control altas, es decir, un bajo tiempo entre solicitudes. Esto da una mayor tasa de refresco en las bases de datos permitiendo actualizaciones mas rápidas.

Finalmente otro punto importante que se evaluó en el período de prueba es el alcance para el manejo de dispositivos en forma inalámbrica. Se llegó a alcanzar 70 metros en el interior de una casa con varios obstáculos en el camino (paredes y puertas) y varias pruebas realizadas en el interior de un edificio dieron un alcance de cuatro pisos con tiempos de respuestas aceptables. Esto demuestra que los componentes de hardware desarrollados no son muy relevantes en la performance global del sistema, dándole mayor protagonismo a la actividad general de la red.

5 EXTENSIONES

Algunas extensiones planificadas, pero no implementadas en esta instancia, son sumamente interesantes. La primera guarda relación con las rutinas y pretende dotar de cierta inteligencia al sistema.

Generación automática de rutinas

Nuestra aplicación permite la definición de rutinas de encendido y apagado de los dispositivos del hogar. La configuración de estas rutinas puede hacerse tanto localmente como en forma remota, como ha sido descripto anteriormente. Sin embargo, el sistema bien podría observar el ciclo de vida de varios dispositivos durante un período de tiempo razonable de actividad normal en el hogar y generar en forma automática rutinas que permiten simular estas actividades. Por ejemplo, la luz exterior de la puerta de entrada de una casa suele usarse únicamente a la noche y el uso del televisor es más común en los períodos denominados *prime-time*. En la habitación de un adolescente, el reproductor de música puede que nunca se utilice a la mañana muy temprano pero sí en horarios nocturnos. Esto por supuesto depende de las rutinas diarias de la familia o del dueño de casa, y sin duda presentará pequeñas variaciones de un día a otro. Pero en términos generales la

utilización de los electrodomésticos y otros aparatos eléctricos del hogar siguen cierto patrón con el correr de los meses. Puede incluirse un módulo de *Aprendizaje para la Generación de Rutinas*.

Existe un trabajo previo realizado, en el cual se desarrolló una aplicación que precisamente se basa en buscar distintos patrones en una base de datos e indicar posibles resultados ante determinados eventos. Esto es lo que se denomina Data Mining o Minería de Datos.

Esta aplicación se adapta a cualquier tipo de problema o inquietud que se quiera conocer por este proceso de patrones. Particularmente fue desarrollada en Java y se utilizaron las librerías de *Weka* [10], para resolver las búsquedas de distintos patrones en la Base de Datos. *Weka* es una colección de algoritmos de aprendizaje automatizado para tareas de Data Mining. Estos algoritmos pueden ser aplicados directamente al conjunto de datos o pueden ser llamados desde su propio código Java. *Weka* contiene herramientas para el pre-procesamiento de datos, clasificación, clusterizacion, reglas de asociación y visualización, y es además una buena herramienta para el desarrollo de nuevos esquemas de aprendizaje automatizado.

La integración de esta aplicación se puede resolver fácilmente si cada acción sobre algún dispositivo es almacenada en una base de datos o archivo de *log*. Se precisar conocer el dispositivo accionado y el día y hora del evento, tanto para las acciones realizadas manualmente como también para aquellas accionadas desde la computadora del usuario. Una vez que tenemos el *log* generado se puede invocar a esta aplicación basada en Weka para que determine, dado un dispositivo o una lista de éstos, en qué momentos fueron accionados generalmente. Luego, en base a una búsqueda en el *log* descripto anteriormente, se determina en qué horarios se realiza comúnmente esta acción, sugiriendo de esta manera o proponiendo una rutina con esos dispositivos y en ese horario. El mismo caso puede ser pensado en el otro sentido, determinar por ejemplo qué dispositivos son accionados en determinada hora durante los días de semana.

Se podría integrar esta funcionalidad por intermedio de un Web Service, que debería encargarse de la lógica para interpretar el resultado de la aplicación de Data Mining y lo traduzca al formato entendido por la aplicación que controla los dispositivos. Este Web Service podría ser consultado cada cierto periodo, como por ejemplo una vez a la semana o una vez al mes. Luego los resultados de este servicio se le muestran al usuario como sugerencias de posibles rutinas que podría agregar al sistema cuya acción corresponde con lo que el usuario realiza diariamente.

Este módulo pertenece a la aplicación local y para su funcionamiento requiere indefectiblemente la capacidad de poder detectar y recolectar el estado real de los dispositivos. La implementación obtenida hasta el momento sólo envía instrucciones hacia los dispositivos, pero no detecta en forma autónoma y directa el estado actual de éstos. La idea es que el puerto paralelo se use en forma bidireccional y de esta forma se puede realizar un monitoreo de los dispositivos que se accionan manualmente. Las aplicaciones de software fueron diseñadas y organizadas pensando en estas expansiones.

Manipulación telefónica de los dispositivos

Otra extensión considerada para una etapa posterior es la posibilidad de controlar la casa por intermedio del teléfono, ya sea la conexión fija de la casa o por intermedio de un teléfono celular que disponga de acceso a Internet. Para la primera posibilidad, es posible añadir en paralelo a la línea telefónica un módulo DTMF, el cual permite la decodificación de los tonos que emite un teléfono, actúa de contestador cuando no hay nadie en la casa, y una vez que atiende la llamada, se queda escuchando instrucciones por un tiempo determinado. Las instrucciones le son enviadas presionando los distintos números del teléfono de quien realizó la llamada y de esta forma se puede dejar asociado cada número para realizar ciertas actividades preestablecidas. Esta implementación es muy práctica ya que con solo realizar una llamada al teléfono de la casa se puede tener control de sus dispositivos. La otra posibilidad que se está considerando actualmente es permitir el acceso al control de la casa por intermedio de un teléfono celular con acceso a Internet. El funcionamiento

básicamente consiste en publicar una versión WAP del servidor que ofrezca los mismos servicios que dispone la aplicación que se maneja en la casa.

6 TRABAJOS RELACIONADOS

Varios autores proponen soluciones domóticas enfocadas en aspectos puntuales. En este trabajo el énfasis es puesto en la universalidad de las tecnologías elegidas en vistas a la facilidad de uso e implementación. Un trabajo interesante es realizado por Marco Aiello [1], donde detalla distintos escenarios en los que se han desarrollado sistemas domóticos. De acuerdo a la clasificación dada por Aiello, nuestra implementación es una instancia del Escenario Domótico *S3* (*open-server hierarchy*) , brindando muy buena heterogeneidad y una aceptable escalabilidad. El único riesgo de este escenario es que al ser centralizado puede volverse un cuello de botella la utilización de los servicios web. Es importante destacar que nuestra implementación también admite un uso fuera de línea (*offline*) de los beneficios del control de dispositivos, por lo que también corresponde a un Escenario Domótico *S1* según indicado en [1]. Otro trabajo relevante es el realizado por Araujo *et al* [5], en el cual se propone una implementación para el manejo de dispositivos de manera inalámbrica. El sistema realizado por nosotros presenta similitudes con el *Radio–Hotel* mencionado en [5]: el aire acondicionado de la habitación es encendido o apagado manualmente pero también se permite cambiar el estado desde la Recepción del Hotel. Nuestro sistema puede funcionar perfectamente en una Intranet, con o sin servidor central. Una diferencia importante es que la propuesta de Araujo *et al* captura el estado de los dispositivos o sensores y actúa automáticamente de acuerdo al estado que se devuelva en la lectura, algo ausente en nuestra propuesta dado el costo de algunos sensores. También Vittorio Miori *et al* desarrollaron un trabajo [3], en el cual plantean la utilización de *Web Services* como forma de inter-relacionar diferentes protocolos de comunicación o diferentes tipos de redes. En ese trabajo se propone que un administrador interprete el resultado del servicio web y se lo transmita a la arquitectura subyacente.

Nuestro trabajo, al utilizar servicios web, se adapta perfectamente a esta propuesta, ya que es posible manejar dispositivos que utilicen JINI o Komex, creando el controlador correspondiente sin ninguna otra modificación al sistema centralizado. En [4] Bonino y Garbo presentan una interesante aplicación de control de dispositivos para ser utilizada por personas con discapacidades motrices. Puede manipularse por medio del movimiento de los ojos o la cabeza, siendo ésta su principal característica que rige el diseño general. Lo interesante en relación a nuestro trabajo es que contempla la interoperabilidad con diferentes redes domóticas existentes, como Bticino MyHome [8].

7 CONCLUSIONES

Los sistemas domóticos son sistemas inteligentes para casas y departamentos, que integran y controlan las comunicaciones, la seguridad, ambientación climática, y los electrodomésticos. Esto genera beneficios en cuanto a confort, seguridad y ahorro de energía. En este trabajo se presentó la implementación de un sistema de control de dispositivos hogareños utilizando servicios web y una conexión a Internet doméstica tradicional. Los dispositivos puede conectarse a una computadora en la vivienda y ser manipulados y programados *in situ*, o remotamente accediendo a un servidor web.

La interfaz visual de este acceso remoto está implementada utilizando AJAX para optimizar la operatoria general. La arquitectura basada en servicios web permite la independencia entre la aplicación local que controla los dispositivos del hogar y el servidor que permite el acceso remoto.

Para la conexión física de los dispositivos, se construyó un módulo de hardware al cual se le conecta la salida del puerto paralelo y en su interior posee relés para manejar aquellos dispositivos que se encuentren cerca de la computadora de forma cableada, y también contiene el emisor con el respectivo codificador para el encendido o apagado de los dispositivos inalámbricos.

REFERENCIAS BIBLIOGRAFICAS

[1] *The Role of Web Services at Home.* Marco Aiello. Proceedings of the Advanced International Conference on Internet and Web Applications and Services AICT/ICIW 2006. IEEE Computer.

[2] *Seguridad en Redes y Criptografía.* Dante I. González Sánchez. Tesis de Maestría. Instituto Tecnológico y de Estudios Superiores de Monterrey. 2004.

[3] *DomoNet: a framework and a prototype for interoperability of domotic middlewares based on XML and Web Services.* Miori, Tarrini, Manca y Tolomei. International Conference on Consumer Electronics, 2006 ICCE '06. pp. 117-118. 2006 Digest of Technical Papers-IEE.

[4] *An accessible control application for interacting with domotic environments through Gaze and Head Movements*. Dario Bonino, Alessandro Garbo. First International Conference on Ambient Intelligence Developments, September 2006, Sophia-Antipolis, pp.11-27. Ed. Springer-Verlag, ISBN-10: 2-287-47469-2

[5] *Domotic Platform Based on Multipurpose Wireless Technology with Distributed Processing Capabilities*. Araujo, Fraga, Moya y Nieto-Taladriz. 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. pp 3003-3007.

[6] *Web Services Architecture.* W3C Working Group. Editado por David Booth, Hugo Haas et al. Artículo on-line. http://www.w3.org/TR/ws-arch/.

[7] *The Networked Home: An Analysis of Current Developments and Future Trends*. Alladi Venkatesh, Erik Kruse, Eric Chuan-Fong Shih. Cognition, Technology and Work, 5 (1), 23-32.

[8] Bticino MyHome System. http://www.bticino-myhome.it

[9] *AJAX: a new approach to web applications.* Jesse James Garret. Adaptive Path Essay. Febrero de 2005. Artículo on-line. http://www.adaptivepath.com/ideas/essays/archives/000385.php

[10] Weka: Data Mining Software. http://www.cs.waikato.ac.nz/ml/weka/

Usando ATL en la Transformación de Modelos Multidimensionales Temporales

Carlos Neil [1]
carlos.neil@vaneduc.edu.ar

Martin Baez [2]
mbaez@lifia.info.unlp.edu.ar

Claudia Pons [1]
cpons@info.unlp.edu.ar

[1] Facultad de Tecnología Informática
Universidad Abierta Interamericana
Buenos Aires, Argentina

[2] Facultad de Informática
Universidad Nacional de La Plata
Buenos Aires, Argentina

Abstract

Model-Driven Architecture (MDA) is a widely accepted approach to the complex software systems design. MDA proposes the use of models in every software development phase, from specification and analysis to implementation. Model transformation is the foundation of MDA, starting from a platform independent model, the aim is to achieve more specific models, in each step. According to this philosophy we present a temporal multidimensional design methodology which allows us to define concepts independently of any implementation issue. In the present work our aim is using ATL (Atlas Trasformation Language) to define both the metamodel and the transformation rules for providing a framework to derive a relational logical schema from an abstract temporal data model. Additionally we are designing an Eclipse plug-in for implementing the transformation.

Keyworks: Temporal Multidimensional Model, Model Transformation, MDA, ATL.

Resumen

Model-Driven Architecture (MDA) es un enfoque ampliamente aceptado para el desarrollo de sistemas de software complejos. MDA propone el uso de modelos en todas las fases de desarrollo, desde la especificación y análisis hasta la implementación. La transformación de modelos es la base de MDA; comenzando por un modelo independiente de la plataforma el objetivo es lograr, en cada paso, modelos más específicos. Adhiriendo a esta filosofía, presentamos una metodología para el diseño de un datawarehouse temporal que permite definir los conceptos independientemente de la implementación. En el presente trabajo, nuestro propósito consiste en la definición de metamodelos y reglas de transformación usando ATL (Atlas Trasformation Language) que provean un marco para la derivación de un esquema lógico relacional a partir de un modelo de datos conceptual temporal. Además, estamos diseñando un plug-in en Eclipse para implementar dicha transformación.

Palabras claves: Modelo Multidimensional Temporal, Transformación de Modelos, MDA, ATL.

1 INTRODUCCIÓN

Model-Driven Architecture [17] se ha establecido como una arquitectura para el desarrollo de sistemas informáticos; tiene como objetivo brindar una solución para los cambios, tanto de negocio como de tecnología, permitiendo construir aplicaciones independientes de su posterior implementación; representa un nuevo paradigma en donde se utilizan modelos del sistema, en distintos niveles de abstracción, para guiar todo el proceso de desarrollo. Un tipo particular de aplicación, denominado sistemas OLAP (On-Line Analytical Processing), que analiza datos consolidados en un almacenemiento, denominado datawarehoue, es particularmente apropiado para desarrollarse con este enfoque. Los sistemas OLAP se caracterizan por la realización de consultas sobre estructuras de almacenamiento y, aunque en general, estas estructuras de datos pueden ser complejas, la definicion de sus modelos son relativamente simples y, por lo tanto, también lo será la formalización de la definición de las transformaciones utilizadas en MDA.

La idea clave subyacente en MDA es que, si se trabaja con modelos, se obtendrán importantes beneficios tanto en productividad, portabilidad, interoperatividad y mantenimiento. Podemos dividir el proceso MDA en tres fases; en la primera, se construye un modelo independiente de la plataforma (PIM), éste es un modelo del sistema de alto nivel, independiente de cualquier tecnología; luego, se transforma el modelo anterior a uno o más modelos específicos de la plataforma (PSM), éste modelo es de más bajo nivel que el PIM y describe al sistema de acuerdo con una tecnología de implementación determinada; por último, se genera el código fuente a partir de cada PSM. MDA, además, presenta un modelo independiente de los aspectos computacionales (CIM) que describe al sistema dentro de su ambiente y muestra lo que se espera de él sin exhibir detalles de cómo será construido. El beneficio principal del enfoque MDA es que una vez que se ha desarrollado cada PIM podemos derivar, automáticamente, el resto de los modelos aplicando las correspondientes transformaciones en forma vertical. Sin embargo, pueden aplicarse también transformaciones horizontales; esto es, un modelo fuente se transformará en un modelo destino dentro del mismo nivel de abstracción [19]. La transformación de PIM a PIM se utiliza cuando los modelos son ampliados o especializados durante el proceso de desarrollo, sin necesidad de contar con información dependiente de la plataforma. Una de las más obvias transformaciones es la que se realiza entre el análisis y el diseño, concepto relacionado con el refinamiento de modelos [17].

El datawarehouse es una colección de datos no volátiles, que varían en el tiempo, que están orientados a un tema determinado y que se utiliza para tomar decisiones organizacionales [8]. El modelo multidimensional constituye la base del datawarehouse, en él la información se estructura en hechos y dimensiones; un hecho es un tema de interés para la empresa, se describe mediante atributos denominados atributos de hecho, éstos están contenidos en celdas o puntos en el cubo de datos. Un cubo de datos es una representación multidimensional de datos donde éstos pueden verse desde distintos puntos de vista; está formado por dimensiones, que determinan la granularidad para la representación de hechos y jerarquías, que muestran cómo las instancias de hechos pueden ser agrupadas y seleccionadas para los procesos de toma de decisión [3]. En el datawarehouse el tiempo es una de las dimensiones para el análisis [8], [9] pero éste hace referencia al momento en que se realizó una transacción, no se detalla cómo ni cuándo varían los atributos o interrelaciones involucradas en esas transacciones. La necesidad de registrar valores que permitan evaluar tendencias, variaciones, máximos y mínimos, justifican considerar en el diseño del datawarehouse cómo algunos atributos o interrelaciones pueden variar en el tiempo. Por lo tanto, un esquema multidimensional temporal que incluya, además del hecho principal de análisis, esquemas temporales (que no pertenezcan a la jerarquía) permitirá registrar, además, la variaciones temporales de atributos y/o interrelaciones.

Para la construcción del esquema temporal [22] se adaptó un algoritmo que permite en forma semiautomática construir, a partir de un modelo entidad interrelación, el diseño conceptual de un datawarehouse [9]. Se utilizó una extensión del modelo entidad interrelación, ampliándolo con atributos e interrelaciones temporales y, aplicando un algoritmo recursivo, se construyó el esquema conceptual, unificando en un sólo modelo, tanto el esquema multidimensional como el temporal; este esquema permite registrar y analizar las variaciones temporales así como la realización de consultas sobre la estructura multidimensional. La transformación, de un modelo de datos temporal a un modelo multidimensional temporal, se realizó de manera informal en tres etapas: primero, utilizando el algoritmo recursivo, se creó un grafo de atributos; luego, a partir del grafo de atributos más un conjunto de decisiones de diseño para la determinación de cuáles serán dimensiones, jerarquías y atributos de hecho, se derivó el modelo multidimensional temporal. Por último, se establecieron criterios para derivar un modelo logico relacional.
Aplicando los conceptos de MDA en la construcción de un datawarehouse, identificamos un CIM que especifica los requerimientos desde una perspectiva de negocio; un PIM que lo define desde un punto de vista conceptual, sin tener en cuenta ningún detalle tecnológico específico y uno o más PSM´s que detallan aspectos de diseño en distintas plataformas, por ejemplo, ROLAP (OLAP Relacional), MOLAP (OLAP Multidimensional) u HOLAP (OLAP Hibrido) [16].
En el presente artículo proponemos, dentro del marco de la filosofía MDA, formalizar la transformaciones presentadas; primeramente, una transformación horizontal (de PIM a PIM), del modelo de datos temporal al modelo multidimensional temporal, pasando por un grafo de atributos; luego, una transformacion vertical (de PIM a PSM), a una plataforma relacional. Utilizaremos un metamodelo para cada uno de los modelos propuestos y aplicaremos el lenguaje ATL [12], para formalizar las transformaciones.
El resto del trabajo está estructurado de la siguiente forma: en el capítulo 2 presentamos la transformación informal del modelo de datos al grafo de atributos, del grafo de atributos al modelo multidimensional y de este último al modelo relacional; en el capítulo 3 mostramos los metamodelos de datos, de grafos, multidimensional y relacional y dos transformaciones formales; en el capítulo 4 detallamos los trabajos relacionados, tanto los referidos al diseño conceptual de un datawarehouse temporal como a las propuestas de diseño de un datawarehouse en un ambiente MDA; por último, en el capitulo 5, presentamos la conclusión y los trabajos futuros.

2 TRANSFORMACIONES DESCRITAS INFORMALMENTE

La metodología de transformación del modelo de datos temporal al modelo relacional plantea una serie de pasos, descritos originalmente de manera informal [21], que detallaremos a continuación y que, en resumen, consisten en la aplicación de un algoritmo que tiene como entrada un modelo entidad interrelación temporal y, mediante sucesivas transformaciones obtenemos, primeramente, un modelo multidimensional temporal y, finalmente, un conjunto de tablas relacionales. Presentamos con un ejemplo (Figura 1) cómo, utilizando el algoritmo, transformamos un modelo de datos temporal (Figura 2) en un grafo de atributos (Figura 3); luego, a partir de éste, creamos el modelo multidimensional temporal (Figura 4) y, finalmente, las tablas en el modelo relacional. Para la aplicación del algoritmo recursivo, primero, transformamos el modelo entidad interrelación (Figura 1) a un modelo entidad interrelación temporal[1] (Figura 2). El atributo multivaluado se convertirá en una entidad débil con una interrelación temporal (marcada con T) vinculada a la entidad que poseía el atributo temporal; la interrelación temporal se transformará en una entidad con interrelaciones binarias (marcada con T) vinculadas a las entidades participantes [22]. En los casos en que queramos preservar una futura jerarquía, proponemos mantener las dos interrelaciones

[1] Por razones de espacio, no presentaremos la transformación del modelo de datos al modelo de datos temporal.

(la instantánea y la temporal). En el ejemplo (Figura 2), conservamos la interrelación entre *PROVEEDOR* y *LOCALIDAD*. Con el mismo criterio general utilizado para transformar interrelaciones temporales, transformamos la interrelación *venta* en una entidad *VENTA* (Figura 2).

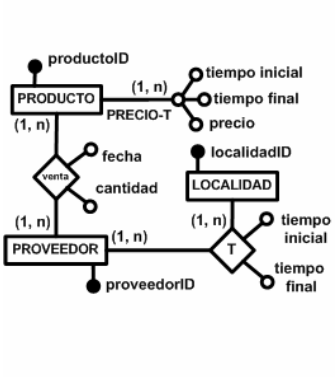


Figura 1. Modelo de Datos

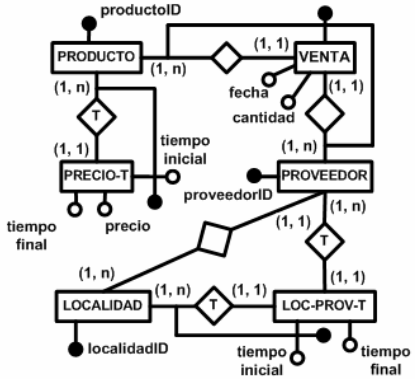


Figura 2. Modelo de Datos Transformado

2.1 Transformación del Modelo de Datos al Grafo de Atributos

Los hechos, como conceptos de interés primario en el proceso de toma de decisión, corresponden a sucesos que ocurren dinámicamente en la realidad, éstos pueden ser representados en el modelo entidad interrelación temporal mediante una entidad E o por medio de una interrelación R n-aria entre entidades E_1... E_n. [9]. Dada un área de interés en un modelo entidad interrelación temporal y una entidad E que pertenece a él, denominamos grafo de atributos al grafo tal que:

- Cada vértice corresponde a un atributo, simple o compuesto del modelo entidad interrelación.
- La raíz corresponde al identificador de E.
- El atributo correspondiente a cada vértice *v*, determina funcionalmente a todos los atributos descendientes de *v*.

Los vértices temporales representan esquemas que tienen como foco de interés la variación de atributos e interrelaciones en función del tiempo. Dado un identifier(E) que indica un conjunto de atributos que identifican a la entidad E, el grafo de atributos (Figura 3) será construido semi automáticamente mediante la aplicación de la siguiente función recursiva modificada de [9]:

```
Function translate (E: Entity): Vertex
{
v = newVertex(E);
// newVertex(E) crea un nuevo vértice,
// conteniendo el nombre y el identificador del  objeto E
for each attribute a ∈ E| a ∉ identifier(E) do
    addChild (v, newVertex(a));
// se agrega un hijo a al vértice v
for each entity G connected to E by relationship R|card-max(E,R)=1 xor R is
temporal do
// se consideran interrelaciones y atributos temporales
    {for each  attribute  b ∈ R do
         addChild (v, newVertex(b));
         addChild (v, translate(G));
    }
return(v);
}
```

Cuando ampliamos el modelo entidad interrelación con aspectos temporales, los atributos y las interrelaciones variantes se transformarán en entidades vinculadas con interrelaciones marcadas con T, del tipo x-a-muchos; por lo tanto, no podrán ser incluidos en la jerarquía para realizar operaciones de agregación. La línea punteada en el grafo de atributos muestra esta particularidad. Por último, probablemente, no todos los atributos representados en el grafo sean de interés para el diseño del datawarehouse. Por tal motivo, éste puede ser modificado para eliminar los niveles de detalles innecesarios [9].

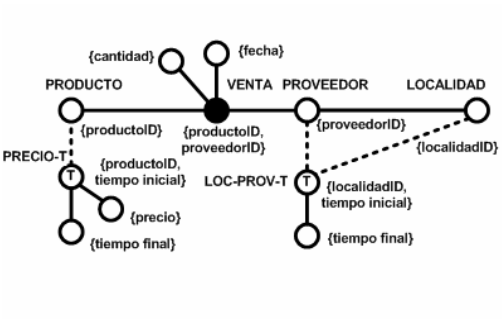


Figura 3. Grafo de Atributos

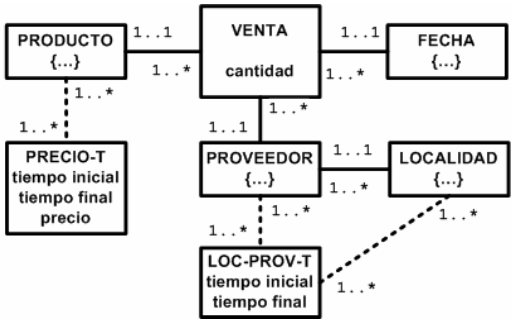


Figura 4. Esquema Multidimensional Temporal

2.2 Transformación del Grafo de Atributos al Modelo Multidimensional

El proceso de transformación del grafo de atributos al modelo multidimensional temporal, esto es, la elección de cuáles vértices del grafo serán atributos de hecho, dimensiones o jerarquías (temporales o no) dependerá de las decisiones del diseñador pero, en general, seguirá el siguiente criterio que utilizaremos en la transformación: la raíz del grafo será el hecho principal; todos los vértices vinculados con la raíz, que no sean identificadores, serán atributos de hecho; los demás atributos vinculados al hecho serán dimensiones; los vértices vinculados a las dimensiones, que no sean identificadores, serán atributos de la dimensión; los demás atributos serán jerarquías (temporales o no) dentro de las dimensiones; todos los atributos vinculados a una jerarquía, si no son identificadores, serán atributos de éstas, sino serán, también, parte de la jerarquía; todas las jerarquías temporales tendrán asociados un rango temporal. El atributo fecha, asociado al hecho, si lo hubiere, será transformado en dimensión. En la figura 4 se muestra el esquema resultante. Los atributos e interrelaciones temporales en el grafo (éstos se vinculan mediante líneas punteadas) precisan de una consideración especial en su transformación al esquema de hecho: éstos no formarán parte de la jerarquía para las operaciones de roll-up y drill down, solamente permitirán evaluar cómo atributos e interrelaciones han variado en el tiempo; constituyen, lo que se denomina, jerarquías no estrictas [26].

2.3 Transformación del Modelo Multidimensional al Modelo Relacional

Por último, a partir del modelo multidimensional temporal obtendremos, aplicando las siguientes reglas de transformacion, un conjunto de tablas en el modelo relacional. El hecho se transformará en tabla; los atributos de hecho, serán columnas de la tabla; la clave primaria estará compuesta por el conjunto de los atributos identificadores; los atributos que forman la clave primaria serán, además, claves foráneas referenciando a cada una de las tablas resultantes de las transformaciones de las dimensiones del hecho. Las dimensiones se transformán en tablas; los atributos de la dimensiones serán columnas de la tabla; la clave primaria estará compuesta por el conjunto de los atributos identificadores; además, cada tabla dimensión tendrá una clave foránea que hará referencia a cada una de las tablas jerarquías vinculadas a la dimensión. Las jerarquías se

transformarán en tablas; los atributos de la jerarquía serán columnas de la tabla; la clave primaria estará compuesta por el conjunto de los atributos identificadores; además, cada tabla jerarquía tendrá una clave foránea que hará referencia a cada una de las tablas jerarquías vinculadas. Las jerarquías temporales se transformarán en tablas. Si la jerarquía temporal deviene de un atributo temporal (isTempAttr = true), tendrá como atributo el tiempo final; la clave primaria será la unión de la clave primaria de la tabla jerarquía vinculada (además, será la clave foránea que hará referencia a dicha tabla jerarquía) más el tiempo inicial. Si la jerarquía temporal deviene de una interrelación temporal (isTempAttr = false), tendrá como atributo el tiempo final y el atributo que es clave primaria de una de las tablas jerarquías vinculadas (además, será la clave foránea que hará referencia a la tabla jerarquía); la clave primaria será la unión de la clave primaria de la otra tabla jerarquía vinculada (además, sera la clave foránea que hará referencia a dicha tabla jerarquía) más el tiempo inicial. A continuacion, se presenta el esquema relacional resultante:

VENTA(productoID(PRODUCTO), proveedorID(PROVEEDOR), fechaID(FECHA), cantidad)
FECHA(fechaID,…)
PRODUCTO(productoID, …)
PROVEEDOR(proveedorID, localidadID(LOCALIDAD),…)
LOCALIDAD(localidadID,…)
PRECIO-T(productoID(PRODUCTO), tiempo-inicial, tiempo-final, precio)
LOC-PROV-T(proveedorID(PROVEEDOR), tiempo-inicial, tiempo-final, localidadID(LOCALIDAD))

3 TRASFORMACIONES DESCRITAS FORMALMENTE

Una regla de transformación de modelos debe definir, evitando cualquier ambigüedad, la relación implícita que existe entre sus partes. MDA no especifica ni prescribe ningún lenguaje para la transformación de modelos. El estándar actualmente establecido por OMG para crear consultas, vistas y transformaciones de modelos es QVT (*Query, Views, Transformations*) [28]. Las transformaciones, en el contexto de QVT se clasifican en relación (*relation*) y función (*mapping*); las relaciones especifican transformaciones multidireccionales, no permiten crear o modificar modelos, pero sí chequear la consistencia entre dos o más modelos relacionados. Las funciones, en cambio, implementan la transformación, es decir, transforma elementos de un dominio en elementos de otro. Se han propuesto varios lenguajes de transformación: BOTL [15]; ATL [12]; Tefkat [14]; Kent Model [1] y también el uso de sentencias OCL [25] para especificar las transformaciones [6], [7]. Todos estos lenguajes asumen que los modelos involucrados en la transformación cuentan con una definición formal de su sintaxis, expresada en términos de metamodelos MOF [20]. En este trabajo hemos utilizado ATL (*Atlas Trasformation Language*) para describir la transformación formal entre los modelos de datos.

3.1 El Lenguaje de Transformación Atlas (ATL)

La transformación de modelos es un componente crítico en MDA; OMG así lo ha reconocido y como respuesta, a través de QVT RFP, han aparecido diversas propuestas. El objetivo de este llamado era definir un lenguaje capaz de expresar consultas, vistas y transformaciones sobre modelos en el contexto de la arquitectura del metamodelado MOF 2.0. La sintaxis abstracta de QVT está definida como un metamodelo MOF; este metamodelo define tres sublenguajes que, en conjunto, forman un lenguaje de transformación híbrido con construcciones imperativas y declarativas denominados: *Relations, Core* y *Operacional Mapping*. Los dos primeros son declarativos, con dos niveles diferentes de abstracción. La documentación de la especificación define la sintaxis textual concreta y la sintaxis abtracta. Además, el lenguaje *Relations* tiene una sintaxis gráfica. El *Operacional Mapping* es un lenguaje imperativo que extiende a los lenguajes *Relations* y *Core.* [13]. ATL comparte las caracteristicas comunes y el mismo conjunto de

requerimientos definidos en QVT RFP. ATL es un framework para administrar transformaciones basadas en modelos. Es un lenguaje mixto, es decir, es una mezcla de construcciones imperativas y declarativas. Un modelo fuente se transforma en un modelo destino mediante una definición de transformación escrita en ATL, que también es un modelo. Los modelos fuente, destino y la definición de la transformación, responden a sus metamodelos respectivos y, a su vez, todos los metamodelos se ajustan a MOF. La transformación de ATL es unidireccional, opera sobre un modelo fuente de sólo lectura y produce un modelo destino de sólo escritura. Durante la ejecución de una transformación, el modelo fuente puede ser navegado pero no cambiado, en cambio el modelo destino no puede ser navegado. Una transformación bidireccional puede implementarse como un par de transformaciones, una para cada dirección [12]. La arquitectura de ATL está compuesta por tres capas; son descritas en niveles decrecientes de complejidad como: *Atlas Model Weaving* (AMW), ATL y ATL *Virtual Machine* (ATL VM). Los programas ATL compilados son ejecutados mediante la ATL VM, que utiliza un conjunto de instrucciones orientadas al modelo. AMW puede ser, opcionalmente, utilizado como un lenguaje de especificación de transformaciones de alto nivel de abtracción [13].
Eclipse [29] es una herramienta que permite integrar diferentes aplicaciones para construir un entorno integrado de desarrolo (IDE). ATL ha implementado un ambiente de desarrollado sobre la plataforma Eclipse; las herramientas disponibles para la transformación de modelos son principalmente dos: el núcleo de funcionalidades ATL, que incluye el motor de transformación y las facilidades de admnistración de modelos. La parte básica de ATL incluye todos los componentes requeridos para configurar y ejecutar transformaciones, en particular, el EMF (*Eclipse Modelling framework*) y MDR (*Meta Data repository*) que permiten, respectivamente, manejar modelos definidos de acuerdo acuerdo a la semántica Ecore y MOF 1.4. Las herramientas basicas de ATL también incluyen una notacion textual simple y la notación *Kernel MetaMetaModel* (KM3) que permite el diseño de metamodelos en forma textual. Las facilidades de manejo de modelos está provista por *Atlas MegaModel Management* (AM3), este módulo tiene como objetivo la administración de los recursos globales en el ambiente de la ingenieria conducida por modelos.

3.2 Metamodelos Usados en la Transformación

Para la especificación de las reglas de transformación es esencial el conocimiento de los metamodelos, tanto de los modelos fuente como de los modelos destino [17]. UML [30] es ampliamente recomendado y aceptado, aunque no especialmente prescripto, como lenguaje de especificación para modelos MDA. A continuación, presentaremos los cuatro metamodelos utilizados para las transformaciones: el metamodelo de datos temporal (Figura 5), el metamodelo del grafo de atributos (Figura 6), el metamodelo multidimensional temporal (Figura 7) y el metamodelo relacional (Figura 8). Todas las clases, excepto las del metamodelo del grafo de atributos, heredan el atributo `name` de una superclase `Named,` no mostrada en los graficos.
A modo de ejemplo, describiremos el metamodelo de datos temporal (figura 5) utilizando KM3; además, ejecutado en la plataforma Eclipse, mostramos la generación del metamodelo en Ecore y su diagrama UML (figura 9).

```
package MMDatosTemporal {
class Relationship {
        attribute isTemp : Boolean;
        attribute name : String;
        reference attributes[0-*] ordered container : Attribute ;
       reference relationshipEnds[2-2] ordered container: RelationshipEnd oppositeOf relationship;
}

class RelationshipEnd {
        attribute multiplicity : Integer;
        attribute rolName : String;
```

```
        attribute name : String;
        reference relationship : Relationship oppositeOf relationshipEnds;
        reference entity : Entity ;
}

class Entity {
        attribute name : String;
        attribute asRoot : Boolean;
        attribute isTemp : Boolean;
        reference attributes[0-*] ordered container : Attribute ;
}

abstract class Attribute {
        attribute name : String;
        attribute isKey : Boolean;
        reference types[0-1] : DataType ;
}

class DescriptiveAttribute extends Attribute {}

class IdentityAttribute extends Attribute {}

class DataType {
        attribute name : String;
}
}
```

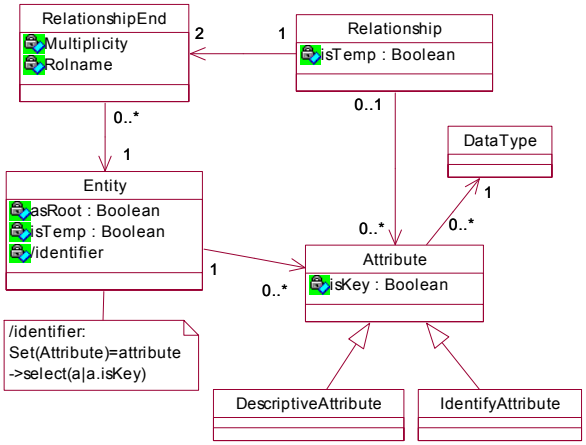


Figura 5. Metamodelo de Datos Temporal

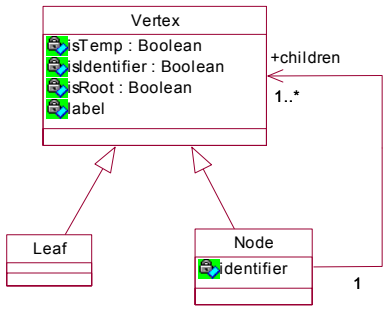


Figura 6. Metamodelo del Grafo de Atributos

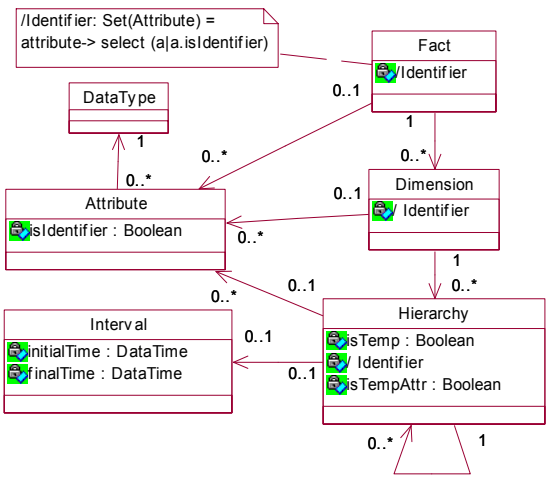


Figura 7. Metamodelo Multidimensional Temporal

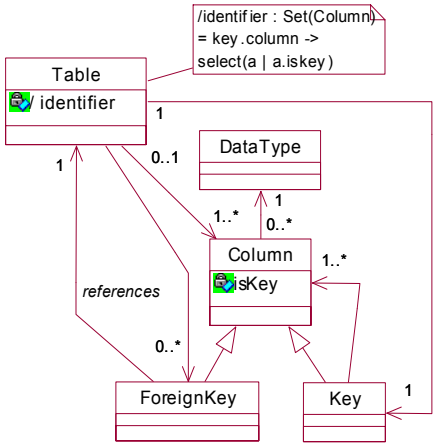


Figura 8. Metamodelo Relacional

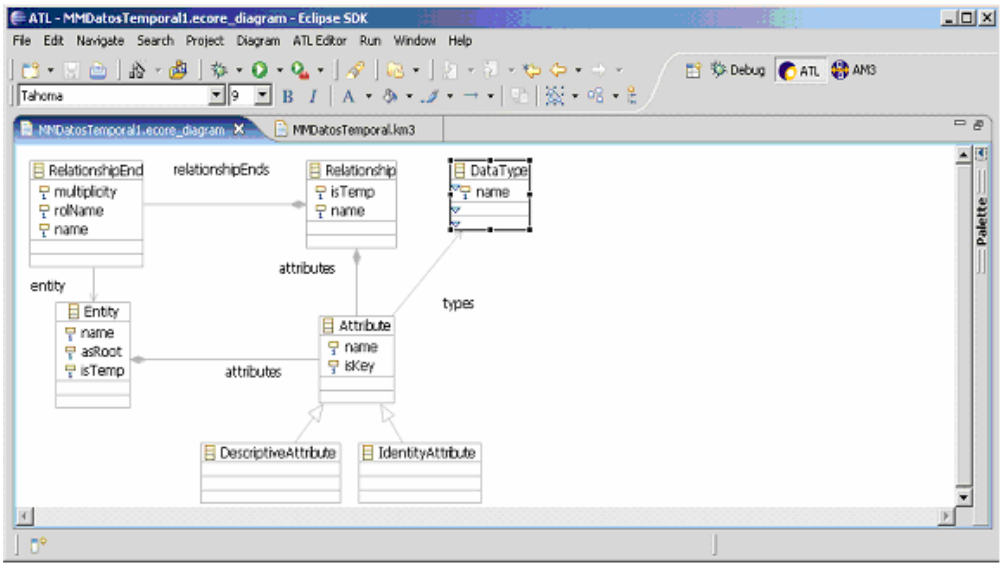


Figura 9. Metamodelo Generado en Eclipse

3.3 Transformaciones Usando ATL

En esta sección, especificaremos formalmente dos transformaciones: la primera, del modelo de datos al grafo de atributos, que ya fue descrita informalmente en la sección 2.1. La segunda, del grafo de atributos al modelo multidimensional, tal como fue detallada en la sección 2.2.

3.3.1 Transformación del Modelo de Datos al Grafo de Atributos

```
module MDatoSTemporales2MGrafo; -- Module Template
create OUT : MMGrafo from IN : MMDatosTemporal;
-- SECCIÓN DE HELPERS ------------------------------------------------------------------------
-- retorna una colección de tuplas(r,g) donde r es una interrelación vinculada a self
-- y g es la entidad del extremo opuesto, con cardinalidad no mayor a 1 o r es temporal

helper context MMDatosTemporal!Entity def: connections():
       Set(TupleType(r : MMDatosTemporal!Relationship, e: MMDatosTemporal!Entity)) =
       self.relationshipEnds
-- Se filtran las relaciones cuya multiplicidad es 1 o son temporales
       -> select (r | r.multiplicity=1 xor r.relationship.isTemp) ->collect(r | r.toTuple());
-- El resultado obtenido se convierte en tupla

helper context MMDatosTemporal!Entity def: identityAttributes():
       Set(MMDatosTemporal!IdentityAttribute) =
       self.attributes -> select (r | r.oclIsTypeOf(MMDatosTemporal!IdentityAttribute));
-- Se filtran las relaciones cuya multiplicidad es 1 o son temporales

helper context MMDatosTemporal!Entity def: descriptiveAttributes():
       Set(MMDatosTemporal!DescriptiveAttribute) =  self.attributes
-- Se filtran las relaciones cuya multiplicidad es 1 o son temporales
       -> select (r | r.oclIsTypeOf(MMDatosTemporal!DescriptiveAttribute));

-- Convierte una RelationshipEnd en una tupla {r,g} donde r es  la relacion y g la entidad
-- del extremo opuesto.
helper context MMDatosTemporal!RelationshipEnd def: toTuple():
       TupleType(r : MMDatosTemporal!Relationship, e: MMDatosTemporal!Entity)=
       Tuple{r=self.relationship,e=self.relationship.entity};

-- Dada una tuple {r,g} donde r es  una relación y g la entidad
-- del extremo opuesto retorna g.
helper context TupleType(r : MMDatosTemporal!Relationship, e: MMDatosTemporal!Entity)
def: getEntity(): MMDatosTemporal!Entity = self.e;

-- Regla called,dado un nombre crea un Nodo hoja del grafo(Leaf).
rule  Descriptive2Leaf(nombre:String) {
to    aLeaf : MMGrafo!Leaf
```

```
do {
        aLeaf.label <- nombre;
        aLeaf.isTemp <- false;
        aLeaf.isIdentifier <- false;
        aLeaf.isRoot <- false;
}}

rule   IdentityAttribute2Identifier {
from   anAttribute : MMDatosTemporal!IdentityAttribute
to      anId:MMGrafo!Identifier(
                name <- anAttribute.name
        )
}

-- Es una regla matched, dada una entidad, la convierte en un nodo raiz del grafo de atributos(Node)
rule  Entity2Vertex {
from  anEntity : MMDatosTemporal!Entity
to    aVertex : MMGrafo!Node ( label <-anEntity.name,isRoot <- anEntity.isRoot,
-- Guardo los identificadores
identifiers <- anEntity.identityAttributes() -> collect( anIdAttr | thisModule.resolveTemp(
anIdAttr,'anId') ),
-- Los atributos descriptivos se convierten en hojas
childrens <- (anEntity.descriptiveAttributes() -> collect(anAttr |
thisModule.resolveTemp(anAttr,'aLeaf')))->
union( anEntity.connections() -> collect(x| thisModule.resolveTemp(x.getEntity())))
}
```

3.3.2 Transformación del Grafo de Atributos al Modelo Multidimensional Temporal

```
module MGrafo2MMultidimensionalTemporal;
create OUT : MMultidimensionalTemporal from IN : MModeloDeGrafo;

helper context MModeloDeGrafo!Node def: verticesNoIdentificadores(): MModeloDeGrafo!Vertex =
           self.childrens -> select (n | not n.isIdentifer);
helper context MMGrafo!Node def: verticesIdentificadores(): MMGrafo!Vertex =
           self.childrens -> select (n | n.isIdentifer);
helper context MMGrafo!Node def: verticesFecha(): MMGrafo!Vertex =
           self.childrens -> select (n | n.label='Fecha');

rule Vertex2Fact {
        from   aRoot : MMGrafo!Node(aRoot.isRoot)
        to      aFact : MMMultidimesionalTemporal!Fact (name<- aRoot.label,
                        dimensions <- Sequence {aRoot.verticesFecha()
                                    -> union(aRoot.verticesIdentificadores())
                                     -> collect(e|thisModule.resolveTemp(e))},
                        attributtes <- Sequence {aRoot.verticesNoIdentificadores()
                                     -> collect(e|thisModule.resolveTemp(e))})}
rule Vertex2Attribute {
        from aLeaf : MMGrafo!Leaf
        to anAtttribute : MMMultidimesionalTemporal!Attributte (name <- aLeaf.label)}

rule Vertex2Dimension {
        from aNode : MMGrafo!Node (not aNode.isRoot)
        to aDimension : MMMultidimesionalTemporal!Dimension ( name<- aNode.label,
            hierarchies <- Sequence{aNode.verticesNoIdentificadores()  ->
            collect(e|thisModule.resolveTemp(e))},attributtes <- Sequence {
           aNode.verticesNoIdentificadores()  -> collect(e|thisModule.resolveTemp(e))})}

rule Vertex2Hierarchy {
        from aNode : MMGrafo!Node
        to aHierarchy : MMMultidimesionalTemporal!Hierarchy (name<- aNode.label)}
```

4 TRABAJOS RELACIONADOS

Se propusieron varias soluciones considerando los aspectos temporales en el datawarehouse: En [5] se presentó un esquema estrella temporal que difiere del tradicional en cuanto al tratamiento del tiempo; mientras éste toma al tiempo como una dimensión más, aquel anula la dimensión tiempo y agrega, como atributos de hecho, el tiempo inicial y el final en cada una de las filas de las tablas del esquema. En [26] se describió, entre las características que un modelo de datawarehouse debería tener, la necesidad de considerar los cambios temporales en los datos y las jerarquías no estrictas.

En [18] se presentó el modelo multidimensional temporal y un lenguaje de consulta temporal, donde se agregan marcas de tiempo en las dimensiones o al nivel de instancias (o ambos) para capturar las variaciones en los atributos de las dimensiones. Entre los trabajos vinculados a la transformación de modelos, en [11] se describió, mediante Meta Object Facility (MOF), la transformación del esquema entidad interrelación al esquema relacional y, utilizando sentencias OCL, se establecieron restricciones en el metamodelo. En [4] se plantearon dos fases para la migración de un sistema relacional a un sistema de base de datos orientado a objetos; en la primera, utiliza reglas de transformación para construir un esquema OO que es semánticamente equivalente al esquema relacional, en la segunda fase ese esquema es usado para generar programas que migren los datos relacionales a una base de datos orientado a objetos. En [10] se estudió la sintaxis y la semántica del modelo entidad interrelación y el modelo de datos relacional y sus transformaciones. En [2] se estudió el problema de la traducción de esquemas entre diferentes modelos de datos, introducen un formalismo teórico gráfico que permite representar uniformemente esquemas y modelos para comparar diferentes modelos de datos y describir el comportamiento de la traducción. En [22] se estableció una conexión formal entre modelos de datos; se utilizaron técnicas de metamodelo basado en MOF para representar la transformación, mediante un algoritmo, del esquema entidad interrelación temporal al modelo multidimensional temporal; se emplearon diagramas de clases MOF y sus correspondientes reglas OCL para establecer restricciones en el modelo y en el metamodelo. En [27] se definió una estrategia para verificar formalmente la corrección de transformaciones entre modelos en el contexto de MDE. Existen trabajos donde, específicamente, se utilizó el enfoque MDA para el diseño de un datawarehouse. En [16] se presentó un método estándar e integrado para el diseño de un datawarehouse; se definió el MMD^2A (MultiDimensional Model Driven Architecture) como un enfoque para la aplicación del marco MDA en el modelado multidimensional. En [31] se propuso un método para el diseño conceptual de un datawarehouse, planteado en tres fases: en la primera se extraen un conjunto de esquemas multidimensionales de las bases de datos operacionales mediante reglas de transformaciones definidas en el marco de MDA, la segunda fase está vinculada con la identificación y la elección de los requisitos del usuario; por último, estos requisitos se usan para seleccionar y refinar los esquemas multidimensionales. En [23] y [24] se presentó, utilizando metamodelos, reglas de transformacion y aplicando el enfoque MDA, una metodología que convierte un modelo entidad interrelacion temporal en un esquema multidimensioneal temporal.

5 CONCLUSIÓN Y TRABAJOS FUTUROS

MDA promueve el uso intensivo de modelos en el proceso de desarrollo, se construyen modelos de los sistemas utilizando primitivas de alto nivel de abstracción; luego, estos modelos se transforman hasta obtener código fuente del sistema final. Inicialmente, se crea un modelo independiente de la plataforma (PIM); luego, se transforma el modelo anterior a uno o más modelos específicos de la plataforma (PSM); por último, se genera el código a partir de cada PSM. En el presente trabajo se desarrolló una metodología semiautomática para generar un esquema relacional de un datawarehouse temporal (ROLAP) a partir de un modelo de datos temporal; primero se presentó un algoritmo recursivo que permitió diseñar un grafo de atributos a partir de un modelo de datos; luego, se estableció informalmente la transformación del árbol de atributos al modelo multidimensional y de éste al esquema relacional; a continuación, se presentaron los metamodelos del modelo de datos temporal, del grafo de atributos, del modelo multidimensional y del relacional. Finalmente, se presentaron transformaciones formales utilizando ATL. En trabajos futuros se desarrollará un plug-ins en la plataforma Eclipse que permita implementer la

transformación del modelo de datos multidimensional para crear un esquema relacional en diferentes Sistemas Administradores de Base de Datos (SABD).

REFERENCIAS

[1] Akehurst D.H., Howells W.G.J., McDonald-Maier K.D. Kent Model Transformation Language Proc. Model Transformations in Practice Workshop, part of MoDELS 2005, Montego Bay, Jamaica. 2005.

[2] Atzeni P, Torlone R., Schema Translation Between Heterogeneous Data Models in a Lattice Framework. 6th IFIP TC-2 Working Conference on Database Semantics (DS-6), Atlanta, Georgia, 1995.

[3] Agrawal R, Gupta A, Sarawagi S., Modeling Multidimensional Databases, Research Report, IBM Almaden Research Center, San Jose, California, 1995.

[4] Behm, A., Geppert, A., Dittrich, K. R. "On the Migration of Relational Schemes and Data Object-Oriented Database System". In Proceedings of Re-Technologies in Information System. Klagenfurt, Austria, Dec 1997.

[5] Bliujute R., Saltenis S., Slivinskas G., and Jensen C. S., Systematic Change Management in Dimensional Data Warehousing. in Proceedings of the Third International Baltic Workshop on Data Bases and Information Systems, Riga, Latvia, 1998.

[6] Cariou, E., Marvie, R., Seinturier, L., & Duchien, L. (2004). OCL for the Specification of Model Transformation Contracts. In J. Bezivin (Eds.), Proceedings of OCL&MDE'2004, OCL and Model Driven Engineering Workshop. Lisbon, Portugal. 2004.

[7] Cariou, E., Marvie, R., Seinturier, L., & Duchien. Model Tranformation Contracts and their Definition in UML and OCL. Technical Report 2004-08, 2004.

[8] Chaudhuri S. and Dayal U., An Overview of Data Warehousing and OLAP Technology, ACM SIGMOD Record 26(1), March 1997.

[9] Golfarelli M., Maio D., Rizzi S., The Dimensional Fact Model: a Conceptual Model for Data Warehouses. International Journal of Cooperative Information Systems, vol 7, n.2&3, 1998.

[10] Gogolla Martin, Lindow Arne, Richters Mark, Ziemann Paul: Metamodel Transformation of Data Models, Workshop in Software Model Engineering, 2002.

[11] Gogolla Martin, Lindow Arne: Transforming Data Models with UML, IOS Press, 2003.

[12] Jouault, F, Kurtev, I: Transforming Models with ATL. In: Proceedings of the Model Transformations in Practice Workshop at MoDELS 2005, Montego Bay, Jamaica.

[13] Jouault, F, and Kurtev, I: On the Architectural Alignment of ATL and QVT**.** In: Proceedings of ACM Symposium on Applied Computing (SAC 06), model transformation track, Dijon, Bourgogne, France. 2006

[14] Lawley Michael, Steel Jim. Practical Declarative Model Transformation with Tefkat, Lecture Notes in Computer Science, Volume 3844, Jan 2006.

[15] Marschall Frank, Braun Meter: Model Transformations for the MDA with BOTL In: Proceedings of the Workshop on Model Driven Architecture: Foundations and Applications, CTIT Technical Report TR-CTIT-03-27, Univeristy of Twente, June 2003.

[16] Mazón Jose Norberto, Trujillo Juan, Serrano Manuel, Piattini Mario: Applying MDA to the Development of Data Warehouses. DOLAP 2005: 57-66.

[17] MDA. Model Driven Architecture. 2004. http://www.omg.org/cgi-bin/doc/formal/03-06-01.

[18] Mendelzon A, Vaisman. A Temporal Query in OLAP. VLDB 2000: 242-253.

[19] Mellor S., Scott K., Uhl A., Weise D. MDA Distilled: Principles of Model-Driven Architecture. Addison-Wesley. 2004.

[20] MOF. Mata Object Facility 1.3. OMG (1999).

[21] Neil Carlos, Ale Juan. A Conceptual Design for Temporal Data Warehouse. 31º JAIIO. Santa Fe. Simposio Argentino de Ingeniería de Software. 2002.

[22] Neil Carlos, Pons Claudia. Formalizing the Model Transformation Using Metamodeling Techniques ASSE Argentinean Symposium on Software Engineering. (33 JAIIO04) September 2004. Cordoba. Argentina.

[23] Neil Carlos, Pons Claudia. Diseño Conceptual de un Datawarehouse Temporal en el Contexto de MDA. XII Congreso Argentino de Ciencias de la Computación. CACIC. San Luis. Argentina. 2006.

[24] Neil Carlos, Pons Claudia. Aplicando MDA al Diseño de un Datawarehouse Temporal. VII Jornada Iberoamericana de Ingeniería de Software e Ingeniería del Conocimiento. Lima, Perú. 2007.

[25] OCL. Object Constraint Language - version 1.5. 2002.

[26] Pedersen T. B., Jensen C. S, Multidimensional Data Modeling for Complex Data. 1998. ICDE 1999.

[27] Pons C. and Garcia D. "An OCL-based Technique for Specifying and Verifying Refinement-oriented Transformations in MDE". Proceedings MoDELS/UML 2006 “Model Driven Engineering Languages and Systems, 9th International Conference, MoDELS 2006, Genoa, Italy, October 2006" LNCS.

[28] QVT Partners. Initial Submission for MOF 2.0 Query/Views/Transformations RFP. Version 1.0 (2003.03.03).

[29] The Eclipse Project. http://www.eclipse.org/

[30] UML 2.0 Infrastructure Specification, OMG document ptc/03-09-15, 2003.

[31] Zepeda Leopoldo, Celma Matilde: Aplicando MDA al Diseño Conceptual de Almacenes de Datos. JIISIC 2006: 271-278.

VALIDACIÓN AUTOMÁTICA DE SISTEMAS

H, Merlino[1,2], **E. Fernández** [2,1], **D. Rodríguez**[2], **P. Britos** [2,1], **R. García-Martínez**[2,1]

[1]Laboratorio de Sistemas Inteligentes, Facultad de Ingeniería, Universidad de Buenos Aires.
Paseo Colón 850 4to Piso. Ala Sur.
(1063) Capital Federal, ARGENTINA.
[2]Centro de Ingeniería de Software Ingeniería del Conocimiento. Escuela de Postgrado. ITBA
25 de Mayo 444 – 6to. Piso
Capital Federal, República Argentina
{hmerlino, rgm}@itba.edu.ar

Resumen

En el presente reporte se hace una introducción a la validación automática de sistemas, se detallan un conjunto de fundamentos sobre la importancia de la validación; a continuación se citan diferentes autores y sus propuestas sobre automatización; luego de esto se hace una reseña sobre algunas herramientas para realizar el proceso y por ultimo a modo de conclusión se da la visión del autor sobre la validación automática de sistemas.

Palabras Claves: Validación automática, automatización.

Abstract

This report describes an introduction to automatic validation systems, the fundamentals about validation's importance are detailed; first there are references about authors, its proposals and automation; secondly there is a summary about some tools required for the process in conclusion it is given the author's vision about.

Keywords: Automatic validation, validation

1. INTRODUCCIÓN

La validación automática de sistemas es el proceso por el cual los requisitos de sistemas son corroborados con poca o nula participación por parte del equipo de proyecto, lo que permite mejorar el producto obtenido. En términos generales esto se logra relacionando cada requisito de sistema a un paquete de pruebas.

La validación automática de software es una práctica que en los últimos tiempos la comunidad de informática le a prestado atención debido a los costos cada vez mas altos que tiene el proceso de complicación-depuración-integración-producción de los sistemas software.

Alguna de las razones por lo cual se hace menester hoy mas que nunca intentar llegar a un proceso de validación automática de sistemas es:

- ✓ *Requerimientos de usuarios cada vez más complejos.*
- ✓ *Diversidad de lenguajes con los que se desarrolla un mismo sistema.*
- ✓ *Diferentes plataformas donde se debe ejecutar un sistema.*
- ✓ *Distintos entornos de programación.*
- ✓ *Constante avance de las tecnologías y métodos de desarrollo.*
- ✓ *Alta rotación de personal.*

2. AUTOMATIZACIÓN DE PROCESOS DE PRUEBA

2.1 Introducción

Fewster [1], cita que la automatización de las pruebas representa un proceso diferente al del proceso de prueba. En consecuencia podemos decir que la calidad de la automatización de las pruebas es independiente de la calidad de la prueba. Para realzar un eficiente proceso de pruebas automatizado se debería determinar un buen proceso de integración de herramientas de prueba y construcción de software. En el siguiente diagrama se representa la evolución en el tiempo de al automatización de los procesos de prueba (Gráfico 1)

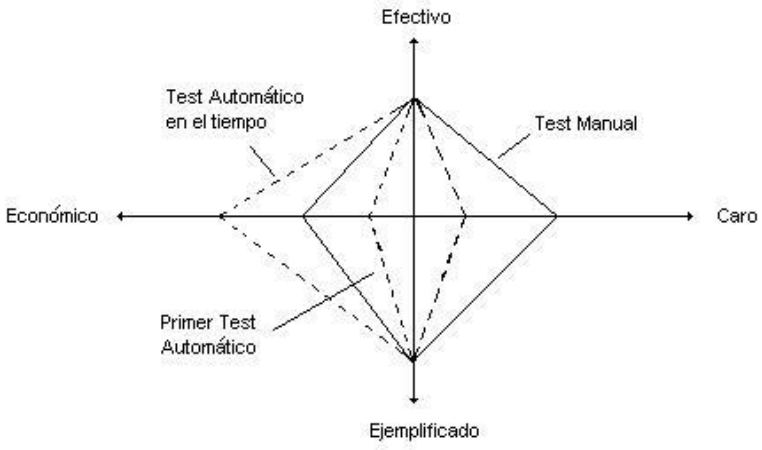


Gráfico 1. Evolución en el tiempo de al automatización

En el Gráfico 1 se observa como a medida que se consigue un grado mayor de automatización el proceso económica viable y efectivo en comparación a un proceso manual de validación.

A continuación se debería poder definir claramente que procesos de validación automatizar, en el presente trabajo y por lo extenso del tema en cuestión; se ha centrado la atención en los temas relacionados con la puesta en producción del software y la validación funcional de un sistema. Se ha dejado de lado la validación de requerimientos, documentación y arquitectura seleccionada, como así también selección de herramientas lenguajes y diseño arquitectónico.

La validación automática no siempre es recomendada, Fewster [1] detalla un conjunto de situaciones en donde la automatización no es más eficiente que las pruebas manuales, estos escenarios son:

- ✓ Las pruebas son ejecutadas rara vez.
- ✓ Cuando el software es muy cambiante.

También Fewster [1], detalla las cualidades de la automatización y los problemas comunes:

- ✓ Cualidades de la automatización
  - Poder ejecutar las pruebas existentes sobre una nueva versión del programa
  - Poder ejecutar mas pruebas en menor tiempo
  - Ciertas pruebas pueden ser muy difíciles de ejecutar manualmente
  - Mejor uso de los recursos
  - Consistencia entre los distintos ciclos de prueba
  - Poder reutilizar las pruebas
  - Incrementar la confianza en las pruebas.
- ✓ Problemas comunes de la automatización de las pruebas
  - Expectativas poco realistas
  - Poca experticia en las pruebas
  - Las expectativas que una prueba automática detectara muchos errores
  - Falso sentido de seguridad
  - Mantenimiento de las pruebas automatizadas
  - Problemas técnicos
  - Problemas de organización

2.2 Normas para realizar pruebas de sistemas

Para que las pruebas manuales o automáticas sean satisfactorias deben seguir algunas normas que han sido numeradas por varios autores, entre los cuales se detalla las dadas por Myers [2], Burstein [3].

Comunes a Myers [2] y Burstein [3]:

- ✓ Un aspecto necesario de los casos de prueba es la definición del resultado esperado.
- ✓ Las pruebas deben ser hechas por un grupo independiente de los desarrolladores
- ✓ Los casos de prueba deberían ser escritos para condiciones de pruebas inválidas e inexistentes, más que para condiciones conocidas y existentes.
- ✓ Los resultados de la prueba deben ser estudiados minuciosamente.
- ✓ Una prueba es buena cuando se logra encontrar defectos.

- ✓ La probabilidad de encontrar errores en una sección del programa es proporcional a la cantidad de errores ya encontrados en esa sección.
- ✓ El proceso de pruebas es una actividad creativa y un desafío intelectual.

Myers [2]:

- ✓ Examinar un programa para ver si este no hace lo que se supone que debería hacer es solo la mitad del camino; la otra mitad es revisar si hace algo que no se espera que haga.
- ✓ No desechar un caso de prueba al menos que el programa halla sido desechado.

Burstein [3]:

- ✓ La validación de software es un proceso en el cual se seleccionan un conjunto de casos de prueba con la intención de encontrar defectos en el programa probado.
- ✓ Las pruebas deben ser repetibles y reutilizables.
- ✓ Las pruebas deben ser planeadas.
- ✓ Las actividades de prueba deben ser integradas al ciclo de vida de desarrollo de sistemas.

2.3 Catalogación de errores comunes

Es necesario reconocer los distintos tipos de patrones de errores mas comunes que se presentan en el desarrollo de sistemas para minimizar sus efectos en la automatización; Jones [4] agrupa a los errores en categorías generales y aporta estadísticas de cada una de ellas y la relación existente entre los usuarios que utilizan los sistemas y los errores reportados.

Fuentes de errores requerimientos erróneos:

- ✓ Diseño defectuoso.
- ✓ Defectos de codificación.
- ✓ Defectos de documentación.
- ✓ Defectos mal arreglados.

Porcentajes de errores en las distintas etapas:

- ✓ Requerimientos 15%
- ✓ Diseño 40%
- ✓ Codificación 30%
- ✓ Documentación 5%
- ✓ Defectos mal arreglados 10 %

Relación que existe entre usuarios y reporte y cantidad de errores:

- ✓ El número de errores reportados es directamente proporcional a la cantidad de usuarios que utiliza el sistema
- ✓ Cuantos más defectos existen menor cantidad de ellos se encuentra.

2.4 Tipos de pruebas

Otro aspecto importante a considerar es el de reconocer los diversos tipos de prueba que se pueden llevar a cabo dentro de un proceso de validación automática. Para ello se citan algunos autores, según Black [5] define un conjunto de pruebas que se deberían realizar:

- ✓ **Prueba de unidad:** su objetivo es evaluar cada paquete o clase de sistemas, el grado de granularidad de la misma depende de lo que se deba probar.
- ✓ **Prueba de Componentes:** se realiza la evaluación de cada subsistema que conforma el sistema en su totalidad.
- ✓ **Prueba Alfa, Beta:** la prueba Alfa es hecha por el equipo de pruebas de sistema, es decir, es interna; la prueba Beta es realizada por usuarios potenciales.
- ✓ **Prueba de integración:** se evaluá si existen errores en la comunicación de los distintos subsistemas que conforman el sistema total.
- ✓ **Prueba de sistema:** se realiza la prueba funcional del sistema, es evaluar si cumple el objetivo para el que fue planeado
- ✓ **Prueba de usabilidad:** se evaluá cuan amigable es la interfaz de usuario.
- ✓ **Prueba piloto:** es una prueba conducida, guiada, en la cual participan los usuarios.

Myers [2] define en términos generales que áreas se deben probar:

- ✓ **Usabilidad:** se debe evaluar la apariencia del sistema, las fuentes, colores y gráficos que son utilizados.
- ✓ **Performance:** se debe probar el tiempo de carga, y cantidad de transacciones simultanees que soporta, en un tiempo de respuesta aceptable por el usuario.
- ✓ **Reglas de negocios:** validar que los procesos de negocios sean cumplidos.
- ✓ **Transacciones:** evaluar finalización y corrección de las transacciones hecha y la posibilidad de volver hacia atrás.
- ✓ **Integridad de Datos:** evaluar si las transacciones hechas mantienen la integridad de los datos en la base de datos.
- ✓ **Disponibilidad del sistema:** evaluar la respuesta ante fallas conocidas.
- ✓ **Arquitectura de red:** evaluar la conectividad y la carga del vínculo.

Además Myers [2] define los tipos de pruebas que son necesarios

- ✓ **Prueba de unidad de modulo:** El propósito de la prueba de unidad es el de comparar la funcionalidad de un modulo con la funcionalidad especificada definida en el modulo; el objetivo es intentar probar que el modulo contradice las especificaciones de diseño.
- ✓ **Prueba Incremental:** Es la prueba de la integración de distintos módulos del sistema a medida que se van realizando.
- ✓ **Pruebas Top-Down:** Se inicia con el primer modulo o el de mayor importancia de un programa y se va incrementados hasta probarlo completamente.
- ✓ **Pruebas Botton-Up:** A la inversa que el anterior.
- ✓ **Pruebas Funcionales:** Se intenta encontrar discrepancias entre lo que hace el programa y las especificaciones externas.
- ✓ **Prueba de Sistema:** Es probar el sistema con respecto a los objetivos originales.
- ✓ **Prueba de facilidades:** Es probar que cada funcionalidad es cumplida.
- ✓ **Prueba de Volumen:** Es probar el sistema con su carga máxima de datos.
- ✓ **Prueba de Carga:** Máximos usuarios trabajando al mismo tiempo.

Si nos centramos en el desarrollo Web deberían incluir [6] un conjunto de pruebas que se deberían agregar a las antes mencionadas:

- ✓ **Performance del Server:** es evaluar la capacidad del servidor Web y de aplicación.
- ✓ **Prueba de Seguridad:** realizar una prueba de penetración de sistema
- ✓ **Prueba de Memoria:** evaluar la cantidad de memoria utilizada por la aplicación para una carga máxima de usuarios.
- ✓ **Prueba de Firewall:** evaluar las reglas del mismo.

En la presente sección se han sentado las bases para la confección de un plan de pruebas el cual contemple la automatización del mismo.

3. PLAN DE PRUEBAS

Antes de realizar la automatización de la validación de sistemas es necesario definir y planificar los pasos a seguir, evaluar riesgos y contingencias del mismo; es por esto se citan las principales características que deberían tener un plan de pruebas; según Black [5] los pasos de un plan de pruebas deberían ser:

- ✓ *Planificar*: Es el proceso de entender el esfuerzo de la prueba, que se divide en: **(a)** contexto de la organización, **(b)** definir y priorizar riesgos, **(c)** estimar esfuerzo, **(d)** desarrollar un plan de tareas.
- ✓ *Preparar*: Es el proceso de especificar los recursos y las pruebas a realizar, esto se divide en: **(a)** capacitar de ser necesario a los recursos, **(b)** diseñar y evaluar el sistema de prueba.
- ✓ *Ejecutar*: Es el proceso de realizar la prueba, que se divide en: **(a)** ejecutar la prueba, **(b)** documentar el resultado.
- ✓ *Evaluar*: Es el proceso de hacer las correcciones para reiniciar el ciclo, esto se divide en: **(a)** adaptar y mejorar, **(b)** documentar errores, **(c)** comunicar la prueba, **(d)** ajustar cambios.

Para Stottlemyer [6] un plan de pruebas tiene como objetivo el desarrollo de una hoja de ruta para la realización de las pruebas de sistema. Este debe ser documentado y aprobado por los participantes del mismo. La estructura puede ser definida en: **(a)** nombre de proyecto de prueba, **(b)** propósito del documento, **(c)** equipo de prueba, **(d)** riesgos, **(e)** alcance de la prueba, **(f)** ambiente de prueba, **(g)** datos de prueba, **(h)** herramientas de prueba, **(i)** documentación, **(j)** seguimientos de problemas.

En el mismo se debería cuantificar: **(a)** tiempo de respuesta del sistema, **(b)** disponibilidad del sistema, **(c)** seguimiento de problemas encontrados, **(d)** definición de los entregables, **(e)** expectativas de la prueba, **(f)** documentación adecuada, **(g)** estrategias para llegar a los objetivos.

Por ultimo da una lista de tareas para una prueba pueda ser realizada con éxito se debería detallar: **(a)** quien hará la prueba, **(b)** porque se debe hacer esta prueba, **(c)** que se debería probar, **(d)** quienes escribirán los Scripts de prueba, **(e)** que desarrolladores participaran de la prueba, **(f)** que usuarios participaran de la prueba, **(g)** cuando se hará la prueba, **(h)** que documentación se entregar al equipo de auditoria, **(i)** quien seguirá los problemas detectados, **(j)** cual el ambiente de prueba.

Según Craig [7] contempla que los aspectos que debe contemplar un plan de pruebas son: **(a)** determinar la prioridad de pruebas y riesgos, **(b)** priorizar que probar, **(c)** determinar hasta donde probar, **(d)** planificar los riegos y las posibles contingencias.

Para Dustin [8] en un plan de pruebas se debe incluir los siguientes ítems: **(a)** cual es el motivo de la automatización, **(b)** detalle del método de selección de herramientas, **(c)** infamación de la prueba a realizar, **(d)** roles y responsabilidades, **(e)** proceso de diseño y desarrollo de la prueba, **(f)** necesidades de administración, **(g)** método de seguimiento de defectos, **(h)** método de seguimiento de la prueba y **(i)** métricas de la prueba.

3. Metodología del plan de pruebas automático

Existen trabajos en los cuales se hace referencia a metodologías para la automatización de la validación de las pruebas de sistemas, entre las mas sólidas se puede nombras a la propuesta en [8] es la denominada Ciclo de Vida Automatizada de las Pruebas (ATLM Automated Testing Life Cycle Methodology) en la misma se propone seis pasos:

- ✓ Decisión de automatizar la prueba
- ✓ Selección y evaluación de herramientas
- ✓ Recolección de información de las pruebas a realizar
- ✓ Planear, diseñar y desarrollar la prueba
- ✓ Ejecutar y administrar la prueba
- ✓ Analizar los resultados y sacar conclusiones.

ATLM cuenta con varias iteraciones para llegar al proceso final de validación automatizado.

Una metodología que no necesariamente debe ser automatizada es la conocida con el nombre de Modelo V [1] en donde se demuestra que actividades de prueba se deben realizar durante el ciclo de desarrollo de sistema (CDS). El factor mas importante a tener en cuenta en el Modelo V es el de identificar cuando escribir las pruebas y cuando ejecutarlas (Gráfico 2). El objetivo del mismo es relacionar que tipo de prueba debe ser realizada para cada etapa del CDS; a medida que se va realizando una nueva etapa del CDS se escribe la prueba relacionada con esa etapa para ser ejecutadas en orden inverso.

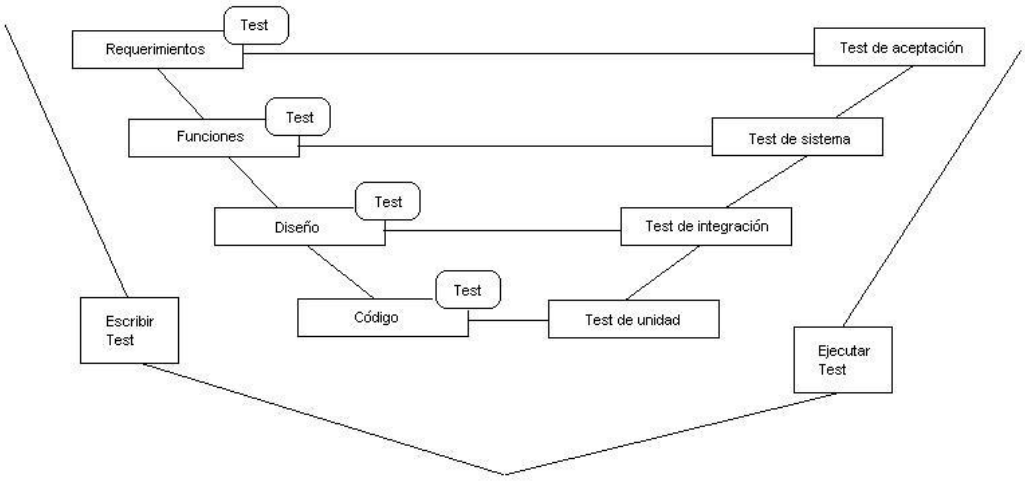


Gráfico 2: Modelo V

A lo largo del ciclo de vida deben ser confeccionadas y ejecutadas distintos tipos de prueba, las cuales son soportadas por distintas herramientas de prueba en el Gráfico 3 se da una breve guiá de las características deseables que debería tener la herramienta seleccionada.

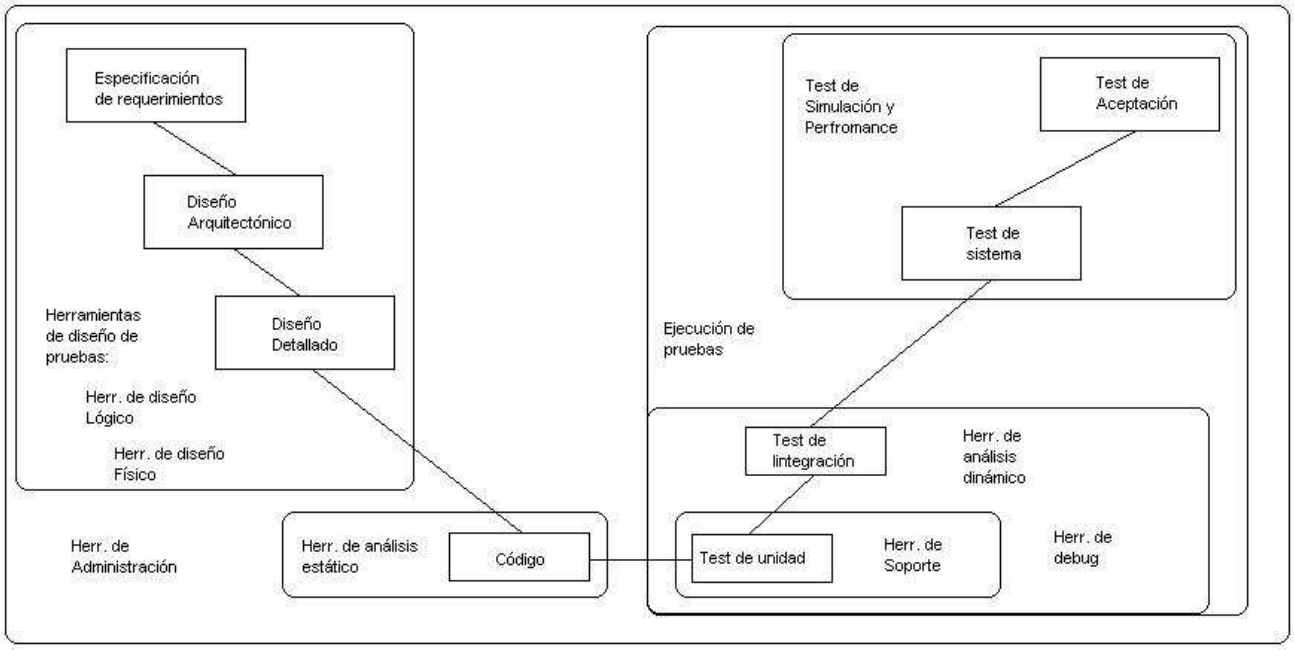


Gráfico 3. Tipo de pruebas

Además se hace hincapié en las actividades que se deben llevar a cabo a lo largo de todo el proceso de desarrollo (Gráfico 4)

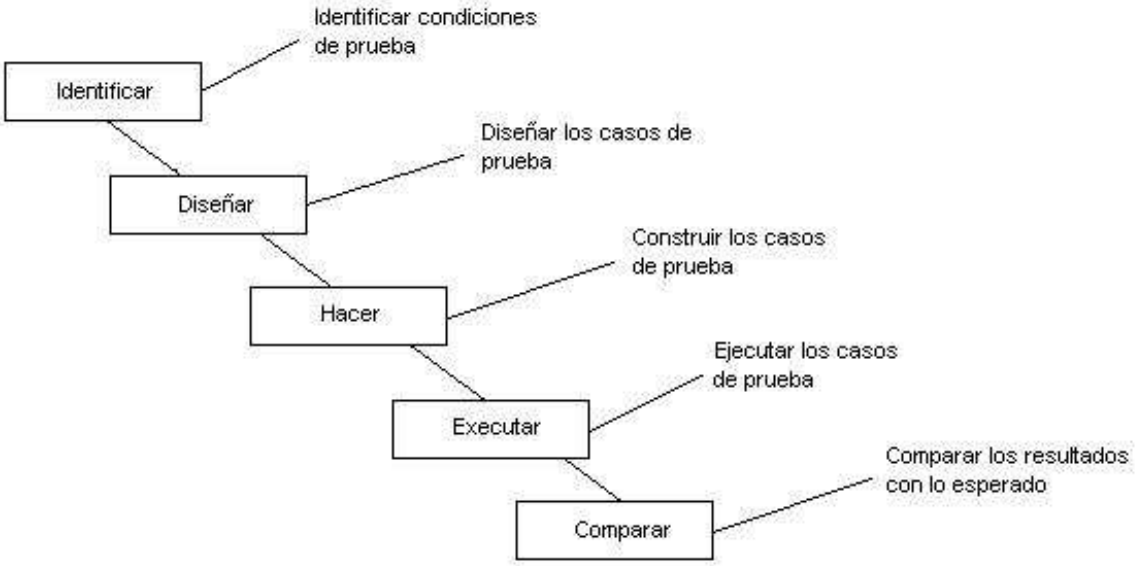


Gráfico 4. Proceso de desarrollo

4. HERRAMIENTAS PARA EL PROCESO DE AUTOMATIZACIÓN Y VALIDACIÓN

La presente lista de herramientas no pretende ser exhaustiva pero si representativa (tabla 1):

Automatización de la puesta en producción	
Producto	**Descripción**
CruiseControl	Framework para realizar el proceso de construcción continua, a este se le pueden adjuntar otras herramientas para complementar el proceso de validación automática, se lo puede considerar como la base principal donde se construirá todo el proceso de automatización.
Anthill	Framework para realizar le proceso de construcción continua y de similares características a CruiseControl.
Ant	Framework para realizar le proceso de construcción, muy configurable.
Validación funcional de sistemas	
Selenium	Framework para probar funcionalidad, imita lo que haría un usuario, principalmente orientado al explorador Mozilla.
Fitnesse	Herramienta que mejora la colaboración en el proceso de desarrollo de software orientada a definir un conjunto de pruebas de aceptación. La principal función de este paquete es la de hacer comprender a los usuarios, programadores y evaluadotes de calidad que hace el software y en forma automática comparar los que realmente hace y lo que debería hacer.
FIT	Herramienta que mejora la colaboración entre los distintos integrantes del desarrollo y sirve para la automatización de las pruebas de aceptación utilizando JUnit para la comprobación
iValidator	Framework basado en XML para la ejecución de escenarios complejos de prueba de funcionalidad de sistemas.
Watir	Framework para probar funcionalidad orientado al explorador de Microsoft
Clirr	Valida funcionalidad actual con versiones anteriores de la maquina virtual de Java.
Validación de unidad de sistemas	
JUnit	Framework de validación de unidad orientado al lenguaje Java.
Checkstyle	Herramienta para la validación de normas de programación.
JDepend	Herramienta para obtener métricas de extensibilidad, usabilidad y mantenibilidad de sistemas.
PathFinder	Herramienta que explora sistemáticamente todos los caminos de ejecución de un programa para detectar abrazos mortales u otras inconsistencias.
FindBugs	Herramienta que detecta patrones comunes de error en Java.
Carga de sistemas	
Grinder	Framework par pruebas de stress de un sistema principalmente orientado a sistemas Web. Este trabaja en la modalidad de Proxy Server con múltiples procesos que imitan ser usuarios

Tabla 1. Herramientas

5. CONCLUSIONES

Se pueden extraer algunas conclusiones sobre el presente trabajo; **(a)** la mejor manera de validar un sistema es relacionar cada requerimiento tanto sea funcional como técnico a un paquete de pruebas; **(b)** la selección de una herramienta de integración continua es la base de todo proceso de validación automática; **(c)** las pruebas no solo deben ser funcionales sino además, como mínimo, de carga de sistemas; **(d)** todo el proceso debe ser detallado en un plan donde se debe contemplar el resultado esperado de cada prueba. Para finalizar se puede mencionar que la validación automática de sistemas no es un proceso trivial, en el cual se debe invertir esfuerzo y recursos para su concreción, el mismo debe ser incluido en el plan de sistemas para su evaluación.

6. APÉNDICE A

6.1. Metodología propuesta

El método propuesto esta organizado en torno a una premisa:

"*Todo los equipos de desarrollo son distintos y enfrentan distintos desafíos*"

Que automatizar y que no, es el desafió al que debe enfrentarse el líder de proyecto, esto se debe pues en el equipo que lidera esta formado con individuos de diferentes intereses y capacidades.

Sobre la construcción del plan de sistemas se recomienda seguir los lineamientos de Stottlemyer [2001], sobre la metodología a utilizar **Modelo V** [1] es una metodología que se adapta en gran medida a la media de los sistemas comúnmente construidos por los ingenieros en software.

Con respecto a la automatización la recomendación es la implementación de las siguientes herramientas, se ha elegido un ambiente de desarrollo Web y desarrollado en Java a modo de ejemplo, utilizar **CruiseControl** como herramienta de integración continua, integrado al control de versiones **Sub-Version**. La carga de las versiones nuevas de software debería ser evaluada por **PathFinder**, **FindBugs** y **Checkstyle** antes de ser ingresadas en el repositorio de código de Sub-Version. Para la construcción de software para los distintos entornos, Prueba, Pre-Producción y Producción se utilizara **Clirr** y **Jdepend**, todos estos procesos son integrados a través de **Ant**.

Para las pruebas funcionales se debería utilizar **Selenium** para validar funcionalidad desde el explorador Mozilla y **Watir** para validar funcionalidad en el explorador de Microsoft. Para las pruebas de carga se utilizaría **Grinder**.

Acerca de la utilización de frameworks como **JUnit** para la prueba de unidad se le recomienda a los programadores utilizarlo, pero no se los obliga, la razón de esto es que escribir pruebas validas se debe tener un alto grado de experticia.

Sobre la validación de sub-módulos el mismo si debería ser realizado con herramientas como JUnit pero las mismas deben ser tareas incluirlas en le plan de pruebas.

REFERENCIAS

[1] Fewster, M. 1999. *Software Test Automation.* Addison-Wesley
[2] Myers, G. 2004. *The art of software testing.* John Wiley & Sons, Inc.
[3] Burstein, I. 2002. *Practical software testing.* Springer Verlag
[4] Jones, C. 1995. *Applied Software Measurement.* Springer Verlag
[5] Black, R. 2005. *Critical Testing Processes*. Addison-Wesley.
[6] Stottlemyer, D. 2001. *Automated Web Testing Toolkit.* John Wiley & Sons, Inc.
[7] Craig, R., Jaskiel, S.. 2002. *Systematic Software Testing*. Artech House.
[8] Dustin, E.. 2002. *Effective Software Testing: 50 Specific Ways to Improve your testing. Editorial*

Verificación Modular de Atomicidad en Bytecode Java Multi-Thread *

Francisco Bavera
Departamento de Computación,
Universidad Nacional de Río Cuarto
Río Cuarto, Argentina
pancho@dc.exa.unrc.edu.ar

Resumen

En este trabajo se presenta una técnica para verificar modularmente atomicidad de programas bytecode Java multi-thread. Los programas deben contar con una especificación referente a los bloqueos y al acceso a los recursos compartidos para realizar la verificación modular. Se presenta la compilación propuesta de programas fuente Java con la especificacion de atomicidad a bytecode Java, con dichas especificaciones incluídas el código compilado. Garantizar atomicidad en programas mult-thread permite, posteriormente, verificar propiedades funcionales más complejas utilizando técnicas para programas secuenciales.

Palabras Clave: Seguridad, Código Móvil Seguro, Concurrencia, Teoría de Lipton

*Este trabajo ha sido realizado en el marco de proyectos subsidiados por la SECyT de la UNRC y por la Agencia Córdoba Ciencia.

1 Introducción

El avance tecnológico ha integrado mecanismos computacionales que comprenden hardware y software a los más diversos artefactos. Cada vez más se busca que estos mecanismos hagan uso de la facilidad de adaptación que significa la posibilidad de variar el software que los integra para dotarlos de nuevas funcionalidades o perfeccionar las existentes. Este fenómeno ha hecho que el problema de poder disponer de Código Móvil Seguro sea cada vez de mayor interés científico e industrial. A su vez el extendido uso de código *multi-thread* en aplicaciones embebidas y en aplicaciones críticas hace necesario contar con un sistema de certificación y verificación que garantice seguridad de dicho código.

El análisis de programas *multi-thread* es difícil y costoso. Esto se debe: (1) al uso, por parte de los *threads*, de objetos compartidos (uso global y concurrente), (2) a la necesidad de utilizar sentencias de sincronización para el uso de objetos compartidos entre *threads*, y (3) al gran incremento del número de posibles trazas de ejecución (por *interleaving* de *threads*). Estas características pueden provocar errores en los programas producidos por no considerar posibles *interleavings* entre *threads* o por usar de forma inadecuada la sincronización.

Los programas *bytecode* Java *multi-thread* son usados por una gran variedad de aplicaciones y tecnologías tal como teléfonos móviles, tarjetas inteligentes y navegadores de internet. Por lo tanto, es necesario y de gran importancia contar con herramientas de verificación para estos programas. Al realizar la verificación sobre *bytecode* no es necesario contar con el código fuente. Además, si el compilador tiene algún bug que introduce algún posible error en el código, dicho error será detectado en la etapa de verificación. Notar que no ocurre lo mismo en el caso de realizar la verificación sobre el código fuente.

Cuando se analizan programas concurrentes es muy importante el modelo de memoria que se utiliza. El Modelo de Memoria de Java (JMM) es una abstracción que describe un conjunto de optimizaciones permitidas. Este modelo abstracto es suficiente para describir optimizaciones de compiladores tales como el reordenamiento de instrucciones. En general, los compiladores (procesadores y caches) se toman ciertas libertades en cuanto al tiempo y orden en la realización de operaciones de memoria. Las técnicas de sincronización son usadas para inducir un cierto orden en la realización de las operaciones de memoria, pero, sin una buena sincronización pueden suceder cosas sorprendentes. Muchos desarrolladores asumen que la sincronización es simplemente una forma de definir *secciones críticas*. Mientras que la exclusión mutua es un elemento de la semántica de sincronización. Existen dos elementos más: la *visibilidad* y el *orden* [4].

La especificación original del JMM, en el capítulo 17 del Java Virtual Machine Specification (JVMS) [8]), es ambigua y tiene serias inconsistencias en su semántica. Esto causa efectos no deseados y/o situaciones no intuitivas para los programadores.

El nuevo JMM, definido por W. Pugh et. al [12], formaliza el modelo de memoria de Java. De esta manera si los programadores conocen (y entienden) el modelo, sus aplicaciones *multi-thread* se comportarán de la manera esperada. Además, según W. Pugh [13], el modelo anterior prohibe la aplicación de muchas optimizaciones de código, mientras que el acceso no sincronizado a objetos inmutables (`final`) puede no ser seguros. El nuevo JMM es parte de Java 5.0 y soluciona las limitaciones que tenía el modelo anterior.

En este trabajo se presenta una técnica para verificar atomicidad de programas *bytecode* Java multi-thread. Los programas a verificar deben contar con una especificación del comportamiento de los bloqueos. Se propone un proceso de compilación de las especificaciones incluídas en el código fuente Java a código *bytecode*. El proceso de compilación genera un archivo `.class` que incluye la especificación. Se podría utilizar un compilador estándar Java (que no realice ninguna optimización) para generar el `.class`. Luego se inserta la especificación de modo tal que el *bytecode* generado pueda ser ejecutado en cualquier maquina virtual Java estándar. Por último, se analiza la relación entre las especificaciones y el JMM. . Este análisis es necesario dado que es muy importante la optimización del código generado. Al optimizar se deben tener en cuenta las restricciones impuestas y analizar las acciones que debe realizar el compilador para no violar ni la especificación ni el modelo de memoria al realizar optimizaciones.

Este trabajo esta estructurado de la siguiente manera, primero se presenta el lenguaje de es-

pecificación usado, sección 2. Dicho lenguaje se basa en una extensión del *Java Modeling Language* (JML) [7] , introducida por Rodrígez et. al [14], para especificar atomicidad. Se presentan, en la subsección 2.2 ,la propuesta del proceso de compilación de las especificaciones JML en el *bytecode* (inserción en el `.class`) y las estructuras definidas para realizar este proceso. En la sección 3, se propone informalmente un método para verificar las especificaciones y realizar la verificación de atomicidad. Luego, en la sección 4, se describe el nuevo modelo de memoria de Java definido por W. Pugh et. al [12]. En la sección 5 se analizan el efecto de las especificaciones sobre las optimizaciones de *bytecode* permitidas por el el nuevo JMM. Por último se presentan las conclusiones y trabajos futuros en la sección 6.

2 El Lenguaje de Especificación

Java Modeling Language (JML) [7] es un lenguaje de especificación formal que permite especificar contratos en programas Java por medio de invariantes de clases, pre y post-condiciones de métodos. Rodrígez et. al [14] presentan una extensión de JML en la cual los constructores provistos permiten especificar atomicidad de métodos y el comportamiento de bloqueos en programas Java *multi-thread*. El concepto de atomicidad es utilizado para evitar los problemas de interferencia interna y externa lo que permite realizar verificaciones de forma modular.

Las especificaciones JML en el código permiten obtener un sistema de certificación/verificación para programas *multi-thread* más efectivo, potente y eficiente. Además, contar con la posibilidad de realizar verificaciones modulares se podría obtener un sistema de certificación/verificación para los contratos de la especificación JML estandar. Para esto se deben compilar las especificaciones JML en el *bytecode* y extender las técnicas de certificación/verificación de contratos a bytecode para obtener un ambiente PCC sobre este *bytecode* anotado. Un componente muy importante de esta línea son los métodos para poder razonar usando las especificaciones JML sobre *bytecode*. En este trabajo solo focalizamos en la verificación de atomicidad.

Las extensiones del JML realizadas por Rodrígez et. al [14] soportan el concepto de atomicidad y permiten la especificación de comportamiento de bloqueos. El concepto de atomicidad (que proviene de la teoría de reducción de Lipton para programas paralelos [9]) es usado para evitar los problemas de interferencia. Este enfoque que introduce concurrencia en JML separa la especificación de los métodos en dos partes: (1) especificación de atomicidad e independencia y (2) especificación de comportamiento secuencial (o funcional).

Interferencia causa problemas que afectan el razonamiento modular del estado de los programas *multi-thread*. Podemos clasificar interferencia en: Interna y Externa. **Interferencia Interna** ocurre cuando un *thread* afecta la ejecución del método corriente de otro *thread* (porque cambió datos que el método puede observar). **Interferencia Externa** ocurre cuando algún otro *thread* realiza un cambio en el estado observable entre la invocación a un método y su ejecución (o entre la salida de un método y el retorno al método que realizó la llamada).

Los programadores pueden usar una gran variedad de mecanismos para asegurar *thread safety* pero la noción central es: no interferencia. Esta es la observación fundamental en la que se basa el trabajo de Rodrígez et al. [14]. Interferencia puede ser evitada usando sincronización. La sincronización evita *interleavings* no deseados y controla el acceso a los datos. Cuando se menciona *thread safety* se esta refiriendo a sincronización, a control de acceso a los datos o una combinación de ambos.

Evitar los problemas de interferencia permite razonar modularmente. El enfoque de Rodrígez et. al [14] se basa en el concepto de **atomicidad** e **independencia**. Una región de código (por ejemplo, el cuerpo de un método) es atómica si las sentencias de esta región son *serializables*. Es decir, para cualquier traza de ejecución de las sentencias de la región (posiblemente intercaladas con sentencias de otros *threads*) hay una traza de ejecución equivalente donde las sentencias de la región son ejecutadas secuencialmente (sin *interleavings* con otros *threads*).

Si una región de código es atómica, entonces es correcto razonar acerca de sus acciones como si estas ocurrieran en un único paso atómico. En otras palabras, se pueden usar técnicas tradicionales de razonamiento secuencial en la región de código atómica. Desde otro punto de vista, para una región de código atómica es correcto considerar solo dos estados: el estado previo (precondición) y

el estado posterior (postcondición). Es decir, cualquier interferencia (*interleavings*) de otros *threads* no produce cambios en su estado, pero este paso atómico puede interferir en la ejecución de otros *threads*.

Una región de código es independiente si todas las sentencias que contiene son independientes. Una sentencia es independiente si puede moverse (reordenarse a derecha e izquierda con respecto a *interleavings* con otros *threads*) sin que se modifique el efecto que produce en el estado del programa cuando se ejecuta. En la teoría de Reducción de Lipton estas sentencias se denominan *both mover* [9].

2.1 Los Constructores de JML para especificar atomicidad

A continuación se presentan los nuevos constructores de JML introducidos por Rodrígez et. al [14] relativos a atomicidad, independencia, bloqueos y otras propiedades específicas de programas *multi-thread*.

2.1.1 Notaciones para Bloqueos

monitors_for <identificador>←<lista de referencias> Esta cláusula permite especificar los bloqueos que protegen el acceso al atributo dado por el identificador. Es decir, significa que todos los bloqueos nombrados en la <lista de referencias> deben ser obtenidos en orden por el *thread* para acceder al atributo <identificador>. Esta <lista de referencias> es una lista de identificadores y atributos separados por comas.

\lockset() Esta expresión retorna el conjunto de objetos que el *thread* corriente mantiene bloqueados en el estado corriente. Por ejemplo, para especificar que el *thread* corriente mantiene bloqueado a o_1 se utiliza la expresión `\lockset().has(`o_1`)`.

locks <lista de referencias> Esta cláusula puede aparecer en la especificación de una precondición y tiene dos propósitos: (1) es una sentencia explícita de los bloqueos que adquiere y libera el método corriente durante su ejecución; (2) establece implícitamente una condición de independencia. Si el método bloquea los objetos especificados entonces la llamada debe garantizar ser independiente. Esto ayuda a realizar verificación modular de atomicidad.

\lock_protected (referencia) Este predicado establece que el objeto referenciado tiene el acceso protegido por algún conjunto de bloqueos (no vacío) y todos estos bloqueos son mantenidos por el *thread* corriente. Hay que notar que es una propiedad muy fuerte ya que que el acceso restringido es sobre el objeto referenciado (no sobre la referencia) y que otros *threads* pueden tener *alias* al objeto pero su acceso está restringido.

2.1.2 Notaciones de Restricción de Heap

rep Este modificador de tipo puede ser usado en la declaración de atributos. Establece que el objeto referenciado por el atributo especificado es parte de la representación de la clase. Es decir, no puede haber ninguna referencia del objeto referenciado por el atributo fuera del objeto de la clase que lo contiene. Esta posibilidad de especificar ausencia de *alias* externos facilita la verificación modular. Este modificador de tipos proviene del Sistema de Tipos *Universe* [11].

readonly Este modificador de tipo puede ser usado en la declaración de atributos. Este modificador establece que el atributo contiene una referencia de solo de lectura (el objeto referenciado no se puede modificar con esa referencia). Las referencias *read-only* no son necesariamente apropiadas por el objeto que contiene al atributo *read-only* (como en el caso de existir *aliasing*). La idea es que solamente la identidad de un objeto *read-only* importa al estado abstracto del objeto. Este modificador de tipos también proviene del Sistema de Tipos *Universe* [11].

\thread_local (referencia) Cláusula que especifica que el objeto referenciado es adueñado por el *thread*. Un objeto `o` es adueñado por un *thread* `t` si `t` es el único que puede acceder a `o`

por alguna cadena de referencias. Esta cláusula es muy usada para verificación modular de atomicidad porque accesos a objetos locales de un *thread* son independientes (no interferentes).

2.1.3 Modificadores de Atomicidad

atomic Este modificador de métodos especifica que cuando un método es invocado en un estado que cumple con su precondición su implementación debe asegurar que la ejecución resultante es serializable. Este modificador resuelve el problema de la interferencia interna ya que estable que todo *interleaving* de otros *threads* durante la ejecución del método no afectan su estado.

2.1.4 Interferencia Externa

\independent Este predicado solo puede ser usado en postcondiciones y especifica que el método cumple la propiedad de ser independiente (todas sus sentencias son independientes). Por ejemplo, accesos a objetos locales al *thread* y acceso a objetos que son protegidos por bloqueos son sentencias independientes, dado que otros *threads* no pueden observar dichos accesos. Es decir, cuando se cumple la precondición la ejecución del método es independiente.

\thread_safe(SR) esta cláusula especifica en la pre o en la postcondición que el objeto `SR` es `\thread_local(SR)` o que es `\locK_protected(SR)`. Esta cláusula se utiliza para evitar interferencia externa deshabilitando el acceso a aquellos objetos que se encuentran en la pre y en la postcondición.

2.2 Compilación de las Especificaciones

En esta sección se describe el formato que se propone para introducir las anotaciones en el archivo `.class`. El formato sigue los lineamientos del formato estándar definido por la JVMS. El formato del archivo `.class` se presenta usando pseudo-estructuras en notación C. Para evitar confusiones (por ejemplo, entre atributos de clases e instancias de clases, etc.) los contenidos de las estructuras que describen el formato del archivo `.class` se denominan *ítems*.

2.3 Información Necesaria para Compilar las Especificaciones

Un archivo `.class` consiste de una secuencia de bytes. 16-bit, 32-bit y 64-bit son construídos por leer dos, cuatro u ocho bytes consecutivos, respectivamente. Los tipos `u1`, `u2` y `u4` representan respectivamente uno, dos o cuatro bytes sin signo.

Las tablas consisten de cero o más *ítems* de tamaño variable; estas son usadas en varias estructuras del archivo `.class`. Si bien se utiliza la sintaxis de arreglos de C para referirse a tablas de *ítems* no es posible traducir un índice de una tabla directamente a un desplazamiento en bytes en la tabla. Esto es consecuencia de que las tablas son secuencias de bytes de tamaño variable Las referencias a estructuras de datos como arreglos pueden ser indexadas como un arreglo, ya que, consisten de cero o más *ítems* continuos de tamaño fijo [8].

Cuando se compila la especificación del método m el compilador usa información de las variables locales (`Local_Variable_table`) y de la tabla que relaciona el número de línea del código fuente con las instrucciones del *bytecode* (`Line_Number_table`). Estos son atributos opcionales de la tabla `attributes` de `Code_attributes` perteneciente a `Method_Info`. Este archivo `.class` puede ser generado por cualquier compilador estándar Java que genere estas dos tablas para todo método que se encuentre en el código. Se considera un compilador estándar a aquel que no realiza ninguna optimización especial en el código generado [1].

Los compiladores pueden definir y generar archivos `.class` con nuevos atributos en las tablas de atributos de estructuras del archivo `class`. Implementaciones dela maquina virtual de Java (JVM) pueden reconocer y utilizar nuevos atributos encontrados en las distintas tablas de atributos. Sin embargo, cualquier atributo no definido como parte de la JVMS no debería afectar la semántica de las clases o interfaces. Por ejemplo, se permite definir nuevos atributos para soportar algún

uso específico [8]. No hay que dejar de mencionar que las implementaciones de la JVM requieren ignorar aquellos atributos que no reconocen. Es decir, los archivos `.class` previstos para cada implementación particular de JVM podrán ser usados por otras implementaciones. Incluso si esas implementaciones no pueden utilizar la información que los archivos `.class` contienen [8].

2.4 Inclusión de las Anotaciones en el Archivo `.Class` Java

2.4.1 Anotaciones en Clases

Las siguientes anotaciones son incluídas como *ítems* en el arreglo de atributos (`attributes[]`) de la estructura `ClassFile` que contiene toda la información de una clase (el `bytecode de una clase`) [8, §4.1]. Cada valor de esta tabla de *ítems* debe ser del tipo `attribute_info` [8, §4.7]. Como ya se mencionó la implementación de una JVM debe ignorar todos los *ítems* en la tabla (de atributos) de la estructura del archivo `Class` que no reconoce. Aquellos *ítems* no definidos en la especificación no afectan a la semántica del archivo `Class` pero proveen información adicional [8, §4.7.1]. En este último caso se encuadran los nuevos *ítems* introducidos. Un tratamiento similar reciben los nuevos *ítems* para métodos, atributos y expresiones.

Ítem `locked_if <predicado>` Esta cláusula permite especificar cuando una instancia de un bloqueo (*lock type*) está en estado de bloqueo. En la figura 1 se puede apreciar la estructura que se define para este *ítem*. Donde,

attribute_name_index es un índice válido en la tabla `constant_pool`. El valor que debe contener la tabla en esa entrada debe ser una estructura `CONSTANT_Utf8_info` que represente el string "Lockedif".

attribute_length contiene la longitud del *ítem* en bytes menos 6 (bytes).

attribute_formula contiene el código de la fórmula que representa el `<predicado>` de la cláusula.

```
Locked_If_attribute {
u2 attribute_name_index;
         u4 attributte_length;
         formula attribute_formula;
}
```

Figura 1: Estructura para el *Ítem* `locked_if`.

2.4.2 Anotaciones en Métodos

Para la cláusula `\atomic` y la cláusula la `\locks` se definió un nuevo *ítem* en el JVM `bytecode` para cada una de las cláusulas. Estos se incluyen en la estructura `method_info` como elementos del arreglo `attributes`. También podrían ser incluídos como un nuevo *ítem* del `JMLMethod_attribute` definido por L. Burdy y M. Pavlova [1, §4.1]. Para ejemplificar se muestra la estructura definida para la cláusula `\locks`:

Ítem `\locks <store_ref_list>` Esta cláusula especifica los bloqueos y desbloqueos que realiza un método. En la figura 2 se puede apreciar la estructura que se define para este *ítem*. Donde,

attribute_name_index es un índice válido en la tabla `constant_pool`. El valor que debe contener la tabla en esa entrada debe ser una estructura `CONSTANT_Utf8_info` que represente el string "Locks".

attribute_length contiene la longitud del *ítem* en bytes menos 6 (bytes).

locks_count cantidad de bloqueos que se encuentran en `<store_ref_list>`.

locks cada entrada en este arreglo representa un bloqueo.

```
Locks_attribute {
u2 attribute_name_index;
u4 attributte_length;
u2 locks_count;
field_info locks[locks_count];
}
```

Figura 2: Estructura para el *Ítem* `\locks`.

2.4.3 Anotaciones en Fields (ítems de clases)

Para las cláusulas `\rep`, `\readonly` y `\monitors_for` se definieron nuevos *ítem*s en el JVM `bytecode` que se incluyen en la estructura `field_info` como elementos del arreglo `attributes`. Como ejemplo se presenta la información generada para la cláusula `\monitors_for`.

Ítem `monitors_for <ident> <-- <store_ref_list>` Esta cláusula especifica que el thread debe mantener (en ese orden) los bloqueos especificados en `<store_ref_list>` para acceder al field `<ident>`. `<ident>` es el field donde se almacena el *ítem* y `<store_ref_list>` es una lista de identificadores, fields, accesos a arreglos, etc. En la figura 3 se puede apreciar la estructura que se define para este *ítem*. Donde,

attribute_name_index es un índice válido en la tabla `constant_pool`. El valor que debe contener la tabla en esa entrada debe ser una estructura `CONSTANT_Utf8_info` que represente el string "Monitorsfor".

attribute_length contiene la longitud del *ítem* en bytes menos 6 (bytes).

locks_count cantidad de bloqueos que se encuentran en `<store_ref_list>`.

locks cada entrada en este arreglo representa un bloqueo.

```
Monitors_For_attribute {
u2 attribute_name_index;
u4 attributte_length;
u2 locks_count;
field_info locks[locks_count];
}
```

Figura 3: Estructura para el *Ítem* `monitors_for`.

2.4.4 Fórmulas y Expresiones

Para compilar las expresiones se extendió la gramática de las expresiones JML presentada por L. Burdy y M. Pavlova [1, §7] para que contemple las cláusulas: `\independent`, `\lockset`, `\thread_local`, `\lock_protected` y `\thread_safe`. El proceso de compilación es similar al presentado por L. Burdy y M. Pavlova [1, §7], solo se debe extender la función de compilación para que contemple estos constructores.

```
JML_Expression ::=  \independent
| \lockset()
| \thread_local( Identifier )
| \lock_protected( Identifier )
| \thread_safe( Identifier )
...
```

Figura 4: Gramática de Expresiones.

3 Verificación Modular de las Especificaciones JML

Para verificar las especificaciones JML de atomicidad sobre *bytecode* se realiza una análisis de flujo de control que explora todos los posibles caminos de ejecución. Es decir, se realiza una ejecución abstracta considerando unicamente las instrucciones que afectan o son afectadas por las especificaciones. Este análisis explota la noción de *independencia*.

A continuación se analizan informalmente las reglas a aplicar para verificar modularmente si un método especificado como **atomic** realmente lo es. Verificar que un método es **atomic** usando la noción de independencia consiste en corroborar que cuando se accede a un objeto este es local al *thread* o que el acceso al objeto está protegido por un conjunto de bloqueos (uno o más bloqueos). En el primer caso, el objeto es local si esta especificado como **rep**, **read_only** o **thread_local**. En el segundo caso, el objeto no solo debe formar parte de una cláusula **monitors_for** y/o **locks** sino que también debe encontrarse (el acceso al objeto) dentro de la cadena de bloqueos establecida. También, el objeto, puede estar especificado como **lock_protected** en la precondición del método.

Para verificar que un atributo declarado **rep** realmente lo es se debe tener la certeza que es imposible que exista una referencia al atributo fuera del *thread* corriente. Esto se puede realizar con *escape analysis*. Verificar que un atributo u objeto es **read_only** o **thread_local** de manera modular (sin conocer el sistema final) es imposible ya que pueden existir *alias* del objeto en otros *threads*. Notar que la existencia de *aliasing* no invalida especificaciones **read_only** y **thread_local**. Con estas especificaciones se podría usar verificaciones dinámicas o verificarlas cuando se integra el *thread* al sistema final. Hay que notar que en algunos casos se puede verificar la clausula **thread_local**, por ejemplo, en aquellos casos donde el objeto especificado es **rep** o esta protegido por algún conjunto de bloqueos.

La especificación **locks** solo implica corroborar que durante la ejecución del método los bloqueos especificados son obtenidos y posteriormente liberados. Verificar la cláusula **monitors_for** no presenta mayores inconvenientes en el análisis del *thread* corriente. Pero, si otros *threads* acceden al objeto deberían hacerlo respetando la misma cadenas de bloqueos. Esto no se puede realizar modularmente. Con lo cual, se puede plantear hacer la verificación antes de la integración del *thread*. Un caso similar sucede con la cláusula **lock_protected** que no especifica cuales son los bloqueos involucrados (solo que existe un conjunto no vacío de bloqueos para un objeto determinado). El problema con esta especificación se encuentra en el punto de verificar cuales son los bloqueos especificados por **lock_protected**. Por ejemplo, para verificar `lock_protected(x.f)` se debería usar la cláusula `monitors_for` para `f` en la clase `x`. Pero con esto se pierde modularidad. Otra alternativa es considerar solo los bloqueos que adquiere el *thread* corriente hasta el punto donde se referencia el objeto especificado por `lock_protected`, los cuales deberían estar especificados en la precondición del método que se está analizando. Pero, esto no garantiza que los bloqueos sean los mismos que los especificados para `f` en la clase `x`. Lo que se verifica en este punto es que todos los métodos que accedan a $x.f$ adquieran siempre la misma cadena de bloqueos (en la misma secuencia).

Para verificar que se cumple la cláusula **thread_safe(O)** hay que chequear que el objeto especificado, **O**, es local al *thread* corriente (**thread_local(O)**) o está protegido por algún conjunto de bloqueos (**lock_protected(O)**). Notar que para verificar la ausencia de interferencia externa también debe verificarse que los objetos que intervienen en la precondición o en la postcondición son **thread_safe**.

Para verificar que se cumple el predicado **independent** se debe chequear que todas las sen-

tencias del método corriente son independientes.

4 Breve Descripción del Nuevo JMM

Un modelo de memoria describe la relación entre las variables de un programa (instancias de atributos, atributos estáticos y elementos de un arreglo) y los detalles de bajo nivel para representar su *binding* con valores, el acceso a ellos y su tiempo de vida sobre la memoria en el sistema de computación real. El modelo de memoria describe como las acciones realizadas por un thread sobre memoria (lecturas y escrituras) afectan las acciones realizadas en memoria por otro thread. Si bien los objetos finalmente son almacenados en memoria, el compilador, el procesador o la cache pueden decidir en que momento mover el valores a la memoria principal.

El modelo de memoria de `Java` permite (al compilador y a la cache) cierta libertad en el orden en el cual se deben mover datos entre registros (o memoria cache) y memoria principal. Pero, el programador puede restringir estas libertades definiendo ciertas reglas de visibilidad (utilizando `sinchronized` o `volatile`). Es decir, en ausencia de sincronización las operaciones en memoria pueden suceder en diferentes ordenes (hasta en los más sorprendentes e inesperados).

El modelo de memoria de Java es una abstracción que describe un conjunto de optimizaciones permitidas. Este modelo abstracto es suficiente para describir optimizaciones de compiladores tal como, por ejemplo, reordenamiento de instrucciones. En general, los compiladores (y también procesadores y memorias caches) se toman ciertas libertades en cuanto al tiempo y orden en la realización de operaciones de memoria. La sincronización es usada para inducir un cierto orden en la realización de las operaciones de memoria. Pero, como es bien conocido en procesos concurrentes, sin una buena sincronización pueden suceder cosas sorprendentes. Muchos desarrolladores asumen que sincronizar es simplemente una forma de definir *secciones críticas*. Mientras que exclusión mutua es un elemento de la semántica de sincronización. Además, existen dos elementos más: la *visibilidad* y el *orden* [4].

El JMM (como estaba especificado originalmente en el capítulo 17 del JVMS [8]) tenía serias inconsistencias en su semántica que permitían ciertas situaciones no deseadas y que no eran intuitivas para los programadores. Además, el JMM no permitía la aplicación de muchas optimizaciones de código, como lo señala W. Pugh [13]. Afortunadamente, fue posible definir un nuevo modelo de memoria para Java. Dicho modelo es consistente con la intuición de los desarrolladores y permite ser utilizado con programas bien sincronizados bajo el modelo anterior [5]. W. Pugh et al. [12] definen y formalizan el modelo de memoria de Java. De esta manera si los programadores conocen (y entienden) el modelo, sus aplicaciones multi-thread se comportarán de la manera esperada.

J. Manson y W. Pugh demuestran que el nuevo JMM permite las siguientes optimizaciones [10]:

- Reordenamiento de Instrucciones. Si se considera un programa P y un programa P' que se obtiene de intercambiar dos sentencias adyacentes x e y de P. Las sentencias x e y pueden ser dos sentencias tal que:
  - reordenar x e y no elimina ninguna relación *happens-before* en ninguna ejecución válida.
  - x e y no son accesos críticos a la misma variable.
  - x e y no son ambos acciones de sincronización.
  - la semántica intra-thread de x e y permite reordenarlas (por ejemplo, x no almacena en un registro leído por y). Si se respeta la semántica intra-thread entonces son legales las optimizaciones de compilación comunes para aplicaciones single-thread.
- *Unrolling/Merging* de Sentencias de Control de Flujo. El compilador puede hacer transformaciones del código que dividan (o unan) el flujo del programa en ejecuciones que sean equivalentes a la ejecución original. Toda forma de *splitting* y *merging* debe preservar la semántica intra-thread.
- Lecturas Especulativas. Una lectura especulativa no puede ser realizada antes que:
  - la última acción de sincronización que obtiene un bloqueo (*acquire*).

Puede Reordenar	Segunda Operación		
Primer Operación	Normal Load Normal Store	Volatile Load MonitorEnter	Volatile Store MonitorExit
Normal Load Normal Store			No
Volatile Load MonitorEnter	No	No	No
Volatile Store MonitorExit		No	No

Donde:

- Normal Loads es: `getfield, getstatic, array load` de fields no volatile.
- Normal Store es: `putfield, putstatic, array store` de fields no volatile.
- Volatile Loads es: `getfield, getstatic` de fields volatile que son accedidos por múltiples *threads*.
- Volatile Stores es: `putfield, putstatic` de fields volatile que son accedidos por múltiples *threads*.
- MonitorEnters bloqueos de objetos accedidos por múltiples *threads* (incluye entradas a métodos sincronizados).
- MonitorExits desbloqueos de objetos accedidos por múltiples *threads* (incluye salidas de métodos sincronizados).

Figura 5: Reordenamientos que se permiten en el nuevo JMM [6].

– la escritura de una variable de la cual se lee.

Las lecturas especulativas anteriores no excluyen a otras posibles optimizaciones que puedan ser permitidas por el modelo. Pero esta demostrado que estas no violan el JMM [10]. Dicha demostración no asevera nada de otras posibles lecturas especulativas.

Para un compilador el JMM consiste principalmente de reglas que deshabilitan el re-ordenamiento de ciertas instrucciones (bloques y accesos a objetos). En la figura 5 se muestran las instrucciones (asociadas a secuencias de *bytecode*) que no pueden ser reordenadas [6].

Las celdas en blanco de la tabla de la figura 5 significan que el reordenamiento es permitido si las instrucciones no son dependientes (según la semántica básica de Java). Por ejemplo, no se puede reordenar un `Normal Load` con un `Normal Store` de la misma posición de memoria. Pero, si se permite el reordenamiento de estas operaciones sobre posiciones distintas de memoria. Las celdas que contienen `No` significan que la primer instrucción no puede ser reordenada con ninguna ocurrencia posterior de la segunda operación.

El nuevo modelo de memoria también da una nueva semántica para `volatile` y `final`. La semántica original de `volatile` garantiza que las lecturas y escrituras de atributos `volatile` debe ser realizadas directamente de memoria principal y que se deben ejecutar en el orden en que se encuentran en el *thread*. Pero, en ella no se especificaba nada en cuanto a la relación entre atributos `volatile` y `no volatile`. Es decir, se permitía realizar cualquier reordenamiento que involucrara una operación sobre un objeto `volatile` y uno `no volatile`. Por lo cual, los atributos `volatile` no podían ser usados como *flags*. El nuevo JMM no permite cualquier reordenamiento entre lecturas y escrituras `volatile` y otras operaciones de memoria (como se puede apreciar en la figura 5).

El nuevo JMM garantiza que los atributos `final` sean visibles a otros *threads* luego de su inicialización. Para lograr esto, no permite que la referencia al objeto este disponible hasta que su constructor no se termine de ejecutar. El modelo de memoria anterior permitía (en ausencia de sincronización) la posibilidad que algún *threads* obtenga el valor por defecto de un atributo `final` y luego en otro instante de tiempo obtenga el valor correcto.

5 Las Especificaciones y el Nuevo JMM

En esta sección, se analiza el efecto de las nuevas cláusulas de JML en el modelo de memoria de Java (JMM). Es interesante analizar el impacto de las especificaciones en las optimizaciones permitidas

por el nuevo JMM.

Las especificaciones `monitor_for`, `lockset`, `lock_protected`, `locks`, `atomic` e `independent` no introducen ninguna información que deba ser tenida en cuenta para realizar optimizaciones permitidas por el JMM. Es decir, estas especificaciones no restringen las optimizaciones permitidas ni habilitan nuevas optimizaciones.

De manera similar ocurre con las cláusulas `rep`, `readonly` y `thread_local`. Es decir, permiten reordenar los objetos asociados a estas especificaciones. Pero, estas últimas cláusulas, pueden ser violadas por otros objetos que referencien a los objetos a que hacen alusión las cláusulas en cuestión. Esto no puede ser verificado modularmente. Entonces, que que sucede en aquellos casos en donde no se cumple la especificación. Por ejemplo, si un objeto especificado como `thread_local` no lo es realmente. Este tipo de casos solo se detectará cuando el *thread* sea integrado a un sistema (el sistema a final donde será ejecutado). Por lo cual se pueden plantear dos alternativas: (1) Realizar una verificación de la integración. Verificar que en el sistema multi-thread cumplen lo especificado, por ejemplo, analizando posibles condiciones de *aliasing* de determinados objetos. (2) Introducir verificaciones dinámicas que den una excepción si no se cumple lo especificado. Por ejemplo, introduciendo código bytecode que realice esta verificación. O modificar la JVM para que dispare una excepción cuando se viole una de las condiciones especificadas.

Como se puede apreciar utilizar estas especificaciones en programas *bytecode* no limitan las optimizaciones permitidas por el JMM. También, se debe notar que estas especificaciones no introducen ninguna inconsistencia con respecto a la especificación del JMM.

6 Conclusiones y Trabajos Futuros

Se presentó una técnica que permite verificar atomicidad de aplicaciones *bytecode* multi-thread. Verificar atomicidad permite posteriormente verificar propiedades funcionales más complejas como si fueran programas secuenciales. Se presentó el proceso de compilación de la especificación a Java *bytecode*. Se analizó la utilización del entorno con el nuevo JMM, concluyendo que la técnica presentada no introduce inconsistencias con respecto al modelo de memoria de Java.

El proceso de compilación presentado permite utilizar cualquier compilador Java (que no realice ninguna optimización), luego se pueden insertar la especificación JML y realizar las optimizaciones que se deseen (mientras respeten el JMM). Las especificaciones no condicionan las optimizaciones permitidas por el modelo de memoria ni las optimizaciones vuelven inconsistentes las especificaciones.

Cualquier JVM (que cumpla con la JVMS) puede utilizar el *bytecode* con la especificación JML. Esto se debe a que si la JVM no reconoce la especificación la descarta. Además las especificaciones no afectan a la semántica del *bytecode*. Si se desea verificar el *bytecode* se puede insertar un módulo (el verificador) antes de la JVM.

El JML extendido utilizado en este trabajo no permite verificar ausencia de *deadlock* y *starvation*. Tampoco permite verificar las consecuencias de integrar un *thread* a un sistema. Los consumidores de código deberían publicar los bloqueos (y el orden en que se deben realizar) que se deben mantener para acceder a cada objeto. También deberían publicar que objetos deben ser `thread_local`. Estos son aspectos que se desean profundizar e integrar al entorno presentado.

También es importante y de gran relevancia realizar casos de estudio complejos.

Se desea estudiar la posibilidad de generar automáticamente las especificaciones con este JML extendido. Pero generar modularmente la información de las cláusulas `thread_local` y `readonly` no es trivial.

También se desea generar un método que permita realizar la verificación y que genere el esquema de prueba para que pueda ser utilizada en un ambiente de código móvil siguiendo las ideas de de la técnica *Proof-Carrying Code* (PCC). Esta técnica exige a un productor de software que entregue su programa ejecutable conjuntamente con una prueba formal de que dicho programa respeta la política de seguridad del receptor. Dicha política de seguridad se formaliza mediante un sistema de axiomas y reglas de inferencia, sobre el cual debe basarse la demostración construida por el productor. El consumidor, por su parte, verifica que la prueba sea válida y que el código recibido

corresponda a la demostración, y sólo ejecuta el código en caso de que ambas respuestas sean positivas.

6.1 Trabajos Relacionados

Verificar, basándose en la Teoría de Reducción de Lipton [9], que un método declarado atómico lo es realmente puede realizarse mediante una gran variedad de caminos. Flanagan y Qadeer [2] utilizan reducción para verificar atomicidad en un sistema de tipos para `Java`. Otro técnica usada para detectar violaciones de atomicidad (usando reducción) consiste en determinar dinámicamente el conjunto de bloqueos que protegen el acceso a los datos compartidos y los objetos locales de los *threads*, técnica utilizada por la herramienta `Atomizer` [3]. Hatcliff et al [14] utilizan *Model Checking*, los resultados experimentales que obtuvieron permiten pensar que es factible realizar modularmente la verificación de atomicidad. También, se puede utilizar analisis de flujo de control y verificar independencia como se desarrolló en este trabajo.

Referencias

[1] Lilian Burdy, Mariela Pavlova. "Specification of java Modeling Language coding into the Java Bytecode" (Draft Version). Reporte Técnico del proyecto Everest, INRIA, Sophia-Antipolis. Febrero, 2005.

[2] C. Flanagan, S. Qadeer. ´´´A Type and Effect System for Atomicity". Proceedings of the ACM Conference Programming Language Design and Implementation. pp. 338-349. 2003.

[3] C. Flanagan, S. Freund. ´´Atomizer: A Dinamic Atomicity Checker for Multithreaded Programs". Proceedings of the ACM Symposium on the Principles of Programming Languages. 2004.

[4] Brian Goetz. "JSR 133 in Public Review". java.net articles. Abril de 2004. http://today.java.net/pub/a/today/2004/04/13/JSR133.html

[5] Brian Goetz. "Fixing the Java Memory Model, Part 1 y Part 2". Java Theory and Practice, IBM developerWorks on Java programming concepts, techniques, and best practices (columna mensual). Febrero-Marzo de 2004. http://www-106.ibm.com/developerworks/java/library/j-jtp03304/

[6] Doug Lea. "The JSR-133 Cookbook for Compiler Writers". http://gee.cs.oswego.edu/dl/jmm/cookbook.html

[7] G. Leavens, E. Poll, C. Clifton, Y. Cheon, C. Ruby, D. Cok, J. Kiniry. "JML: Reference Manual". Departament of Computer Science, Iowa State University. Disponible: www.jmlspecs.org. 2004.

[8] Tim Lindholm, Frank Yellim. "Java Virtual Machine Specification". Segunda Edición, Java Software, Sun Microsystems, Inc., 2004.

[9] R. Lipton. "Reduction: a method of proving propierties of parallel programs". Communications of the ACM 18 717-721. 1975.

[10] Jeremy Manson, William Pugh. "Proof sketch the Manson/Pugh JMM allows reordering, Unrolling/Merging and Speculative Reads". JMM web page (http://www.cs.umd.edu/users/pugh/java/memoryModel). Febrero de 2004.

[11] Peter Müller, A. Poetzsch-Heffter. "A Type System for Alias and Dependency Control". Reporte Técnico 279, Universidd de Hagen. 2001.

[12] Jeremy Manson, William Pugh, Sarita V. Adve. "The Java memory model". POPL 2005. pp 378-391.

[13] William Pugh. "The Java memory model is fatally flawed". Concurrency - Practice and Experience 12(6). pp 445-455. 2000.

[14] Edwin Rodríguez, Matthew B. Dwyer, Cormac Flanagan, John Hatcliff, Gary T. Leavens, Robby. "Extending Sequential Specification Techniques for Modular Specification and Verification of Multi-Threaded Programs". ECOOP 2005.

XM-tree, un nuevo índice para Recuperación de Información en la Web

Claudia Deco, Guillermo Pierángeli, Cristina Bender
Departamento de Sistemas e Informática
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario
(2000) Rosario, Argentina
{deco, bender}@fceia.unr.edu.ar

y

Nora Reyes
Departamento de Informática,
Universidad Nacional de San Luis
(5700), San Luis, Argentina
nreyes@unsl.edu.ar

Abstract

Web Information Retrieval is another problem of searching elements of a set which are closest to a given query under a certain similarity criterion. It is of interest to take advantage of metric spaces in order to solve a search in an effective and efficient way. In this article, we present an extension of the M-tree index, called *XM-tree*, in order to improve search results. This index allows dynamic insertion of new data, reduces search costs using prunings and precalculated distances, and uses a tolerable amount of space, which makes this index apt for the extensive and dynamic Web. The proposed extension indexes Web documents, uses L_2 as indexing distance and L_∞ as similarity criterion to solve queries. We also present experiments validating the results.

Keywords: Metric Spaces, Similarity Searching, M-tree, XM-Tree

Resumen

La Recuperación de Información de la Web es uno más de los problemas de buscar en un conjunto los elementos más cercanos a una consulta dada bajo un cierto criterio de similitud. Es de interés aprovechar las cualidades de los espacios métricos con el objeto de resolver una consulta de manera efectiva y eficiente. En este artículo, se presenta una propuesta de búsqueda utilizando *XM-tree* que es una extensión del índice M-tree. Este índice permite la inserción dinámica de nuevos datos, reduce los costos de búsqueda con distancias precalculadas y podas, y utiliza una cantidad de espacio tolerable, lo que lo hace apto para el extenso y dinámico entorno Web. La extensión propuesta indexa documentos Web, emplea L_2 como distancia de indexado y resuelve las búsquedas aplicando como criterio de similitud la norma L_∞. Se presenta la experimentación que valida los resultados.

Palabras claves: Espacios Métricos, Búsqueda por Similitud, M-tree, XM-Tree

1 INTRODUCCIÓN

La búsqueda es un problema fundamental en las ciencias de la computación, presente en casi cualquier aplicación. La operación de búsqueda ha sido tradicionalmente aplicada a datos estructurados. Con la evolución de las tecnologías de información y comunicación han surgido repositorios de información no estructurada. Este es el caso de la web, que está constituida por millones de páginas. Los nuevos tipos de datos (texto, imágenes, audio, vídeo) son difíciles de estructurar, tanto manual como computacionalmente, lo que restringe los tipos de consultas a realizar. Un entorno de este tipo requiere algoritmos y modelos de búsquedas más generales. Surge así el concepto de *búsqueda por similitud* o *búsqueda por proximidad*, que consiste en recuperar los elementos similares o cercanos al elemento consultado. La similitud se modela con una *función distancia* que satisface la desigualdad triangular y el conjunto de objetos se llama espacio métrico. En algunas aplicaciones el espacio métrico se torna un tipo particular llamado espacio vectorial.

Muchos trabajos apuntan a disminuir el costo de calcular las distancias entre los objetos, y reducir la E/S, construyendo un índice, es decir una estructura de datos para reducir el número de evaluaciones de distancia en el momento de consulta. En [3] se presenta un marco unificado que describe y analiza las soluciones existentes a este problema, y se distinguen dos principales técnicas de indexado: una basada en pivotes y otra basada en particiones de Voronoi. Analizadas las distintas técnicas, se propone realizar una extensión al *M-tree*, para indexar los documentos de la web y acelerar las búsquedas descartando mayor cantidad de objetos irrelevantes.

El resto del trabajo se organiza de la siguiente forma: en la Sección 2 se presentan conceptos básicos de Recuperación de Información y de Espacios Métricos. En la Sección 3 se presentan trabajos relacionados. En la Sección 4 se propone el *XM-Tree* como una extensión al *M-Tree*. La Sección 5 presenta los resultados de la experimentación. Finalmente, se presentan las conclusiones.

2 CONCEPTOS BÁSICOS

2.1 Recuperación de Información

La Recuperación de Información, se define de la siguiente manera: dado un conjunto de documentos y una consulta, determinar el subconjunto de documentos relevante a la consulta [1]. Uno de sus desafíos es determinar la relevancia de un documento con respecto a la consulta formulada por un usuario. El proceso de búsqueda es realizado por un motor de búsqueda, el cual maneja un cuerpo de documentos a través de dos pasos: indexado y recuperación. El indexado produce un índice invertido, que se construye a partir del cuerpo de documentos realizando un mapeo entre cada palabra presente en el cuerpo hacia los documentos que la contienen. La ocurrencia de un término en un documento se llama *posting*. El conjunto de postings asociados a un término se almacena en una *posting list*. En la recuperación, el motor de búsqueda recibe como entrada una consulta formada de una o más palabras. Un motor que utiliza índice invertido revisa qué documentos contienen las palabras ingresadas según el índice, y obtiene las posting lists de cada una. Estas listas se procesan para obtener el conjunto de resultados final, según la semántica de la consulta manejada por el motor. Por ejemplo, los motores booleanos exigen que la consulta se formule utilizando operadores booleanos entre las palabras ingresadas, y aplican las operaciones de conjunto para obtener el conjunto resultado.

El desempeño de un motor de búsqueda se puede evaluar a través de los indicadores *Precisión* y *Recall*. La Precisión es el número de documentos relevantes a la consulta dada dividido el total de documentos recuperados. El Recall es el cociente entre la cantidad de documentos recuperados y el total de documentos relevantes de la colección. Un motor alcanza un buen rendimiento cuando maximiza ambos valores; es decir, recupera la mayoría de los documentos relevantes disponibles en la colección con la menor cantidad de documentos irrelevantes.

Una forma de representar los documentos es el *modelo espacio vectorial*, en el cual los

documentos son representados con vectores de palabras. En estos vectores no se consideran términos muy frecuentes, artículos y pronombres. Además, los verbos conjugados, sustantivos y adjetivos, se reducen a una forma canónica con un proceso de *stemming*. Con el conjunto de palabras resultante se representa el documento como un vector en el espacio euclídeo. Cada término canónico representa un eje en este espacio. Si el término t_i ocurre en el documento d, la i-ésima componente del vector de d vale 1, si no vale 0. Las consultas se representan de la misma manera. La similitud entre una consulta q y un documento d se cuantifica calculando el producto interno entre los vectores de q y d: $sim(q,d) = \sum q_i \times d_i$. Esto permite retornar una secuencia de documentos ordenados por su relevancia a la consulta, ofreciendo un mecanismo de ranking de resultados.

Esta representación no captura la distinta importancia entre un término y otro en un mismo documento. La solución es que cada componente represente el peso del término en el documento. Este peso puede ser el número de ocurrencias, pero esta representación favorece a los documentos más grandes. Esto se resuelve empleando el *esquema TF,* o de frecuencia de términos, el cual consiste en normalizar los vectores dividiendo cada componente por la longitud de los mismos. Así, cada componente d_i del vector vale tf_i, o sea la frecuencia del término t_i en el documento d.

El esquema TF no tiene en cuenta la distribución del término en la colección entera. Esto se soluciona con la noción de frecuencia de documento inversa: $idf_i = log\ (N / n_i)$, donde N es el tamaño de la colección y n_i es el número de documentos donde ocurre el término i-ésimo. En la práctica, se combinan las medidas *tf* e *idf* en el *esquema TF/IDF*: $w_{ij} = tf_{ij}\ x\ idf_i$ donde w_{ij} es el valor que se le asigna a la componente i del documento j.

2.2 Espacios Métricos

Un espacio métrico (X, d) está formado por un universo de objetos válidos X y una función de distancia d: $X \times X \rightarrow R^+$ que mide la semejanza o proximidad entre dos objetos cualesquiera. La función distancia tiene las siguientes propiedades: no negatividad ($d(x, y) \geq 0$), simetría ($d(x, y) = d(y, x)$), reflexividad ($d(x, x) = 0$), positividad estricta ($\forall x, y \in X, x \neq y \Rightarrow d(x, y) > 0$), y satisface la desigualdad triangular ($d(x, z) \leq d(x, y) + d(y, z)$). El subconjunto finito U de X, de tamaño $n = |U|$, es el conjunto donde se realizarán las búsquedas, y es llamado el diccionario o base de datos. Los tipos de consultas de interés en los espacios métricos son la *Consulta por rango* $(q, r)_d$ que recupera todos los elementos que están dentro de la distancia r de q, *o* sea $\{ u \in U \mid d(q, u) \leq r \}$ y la *Consulta por k vecinos más cercanos* $NN_k(q)$ que recupera los k elementos más cercanos a q en U, obteniendo un conjunto $A \subseteq U$ tal que $|A| = k$ y $\forall u \in U, v \in U - A, d(q, u) \leq d(q, v)$. En este trabajo se utiliza la *Consulta por rango estricta* $(q, r)_d$ para recuperar todos los elementos que están dentro de la distancia r de q y no la alcanzan, o sea $\{u \in U \mid d(q, u) < r\}$.

Si los elementos del espacio métrico (X, d) son tuplas de números reales, el par es llamado *espacio vectorial*. Entonces, un espacio vectorial k-dimensional es un espacio métrico particular donde los objetos se identifican con k coordenadas de valores reales $(x_1, ..., x_k)$. En los espacios vectoriales, hay varias funciones de distancias definidas. Las más utilizadas son la familia de distancias L_s, definidas como: $L_s((x_1, ..., x_k), (y_1, ..., y_k)) = (\sum | x_i - y_i |^s)^{1/s}$. La distancia L_1, también llamada *de bloques*, es la suma de las diferencias entre las coordenadas. La distancia L_2, o *distancia euclídea*, corresponde a la noción de distancia espacial. Otra muy utilizada es L_∞, que equivale al límite de L_s cuando s tiende a infinito, y se define como la máxima diferencia entre dos coordenadas de los dos objetos dados: $L_\infty((x_1, ..., x_k), (y_1, ..., y_k)) = \text{máx} \mid x_i - y_i \mid$.

Los índices sobre espacios métricos permiten la recuperación efectiva y eficiente de objetos. Efectiva, porque los resultados tienen un alto grado de exactitud por propiedades del espacio y del índice; y eficiente porque los índices se construyen para reducir el número de cálculos y objetos revisados. Estas dos propiedades son muy deseadas en un motor de búsqueda y hacen de la utilización de espacios métricos una alternativa interesante en los procesos de indexado y búsqueda.

En [3] se presenta un modelo unificado de los algoritmos de indexado de espacios métricos.

Todos los algoritmos de indexado particionan el conjunto U en subconjuntos, y construyen un índice para determinar los subconjuntos candidatos donde los elementos relevantes a la consulta podrían aparecer. Los algoritmos de indexado se dividen en dos clases principales: basados *en pivotes* y de tipo Voronoi (o basados en particiones compactas). Los primeros trabajan tomando k puntos llamados *pivotes* y mapean el espacio métrico sobre R^k usando la distancia L_∞. Los algoritmos tipo Voronoi particionan el espacio métrico en subconjuntos según su cercanía a ciertos puntos llamados *centros*. Los algoritmos basados en pivotes necesitan mucho más espacio para almacenar las clases resultantes de los k pivotes que el que utilizan los de tipo Voronoi para el mismo número de particiones. Cuando los algoritmos basados en pivotes cuentan con la memoria necesaria y usan el número óptimo de pivotes, la cota inferior del costo de búsqueda es mejor que la cota de los de tipo Voronoi. Esta propiedad se desvanece a medida que la dimensión intrínseca del espacio métrico crece, es decir, cuando es más difícil buscar en dicho espacio.

La web es un repositorio de tamaño continuamente creciente. Como en este trabajo se pretende discriminar las páginas por su contenido, se plantea representarlas con vectores de sus palabras representativas, lo que presupone el manejo de vectores con miles de componentes. Esto no permite asegurar que la dimensión intrínseca del espacio métrico será alta o no, pero sí que soluciones para espacios vectoriales no son aplicables. Considerando la incertidumbre sobre la dimensión intrínseca del espacio, los algoritmos de tipo Voronoi aparecen como más aptos. Además, el ambiente de la web es muy dinámico, continuamente se están agregando, modificando y borrando páginas. Por esto, el índice propuesto permite la inserción de datos. En este trabajo se optó por extender el algoritmo de indexado M-tree, dado que éste mejora algoritmos anteriores, permite inserciones y el marcado en las hojas de un elemento como eliminado.

2.3 M-tree

El método de acceso *M-tree* [4] es propuesto para organizar y buscar en grandes conjuntos de datos. Los experimentos muestran un comportamiento razonablemente bueno en espacios de alta dimensión y conjuntos de datos de tamaño creciente. El M-tree es un árbol paginado, dinámico y balanceado, que permite insertar y borrar datos de manera dinámica*, manteniéndose balanceado sin la necesidad de tener que hacer reorganizaciones periódicas. Los nodos del M-tree son de tamaño fijo y permiten un número máximo M de nodos hijos. Los nodos hoja almacenan todos los objetos indexados de la base de datos y los nodos internos, llamados *objetos ruteadores*, mantienen objetos de la base de datos con un rol de ruteo asignado por el algoritmo de promoción específico.

Cada nodo N corresponde a una región del espacio métrico indexado (X, d), definida como $Reg(N) = \{ O \in X \mid d(O_r, O) \leq r(O_r) \}$, donde O_r es el objeto ruteador del nodo N y $r(O_r)$ es el llamado *radio de cobertura*. Por definición, todos los objetos en el subárbol con raíz en N están dentro de la región $Reg(N)$. Si N es un *nodo interno*, mantiene como entradas el objeto ruteador O_r y el radio $r(O_r)$, también tiene un puntero asociado, $ptr(T(O_r))$, que referencia la raíz del subárbol, denotado $T(O_r)$ y llamado *árbol de cobertura*. Finalmente contiene una distancia a $P(O_r)$, su objeto padre, es decir el objeto ruteador que tiene a O_r como una de sus entradas. Esta distancia no está definida en la raíz del M-tree. Los *nodos hoja* son similares a los objetos ruteadores, pero el radio de cobertura no es necesario y un puntero almacena el identificador del objeto en la base de datos.

En [4] se presentan dos algoritmos de búsqueda por similitud: consulta por rango y consulta de k vecinos más cercanos. El objetivo de ambos es reducir la cantidad de nodos accedidos y la cantidad de distancias calculadas, utilizando las distancias precalculadas almacenadas en los nodos del M-tree, constituida por $d(O_r,P(O_r))$ y $r(O_r)$, aplicando la desigualdad triangular. Si bien tratan las consultas por rango como no estrictas, aquí se utilizan consultas por rango estrictas, que son un caso particular de las anteriores y sólo requieren pequeñas modificaciones en su tratamiento. La consulta

* La eliminación se realiza en las hojas, marcando el elemento como eliminado, por si se usa como objeto ruteador.

estricta $(Q, r(Q))_d$ selecciona todos los objetos de la base de datos tal que $d(O_j, Q) < r(Q)$.

Para determinar los nodos relevantes a la consulta utiliza el algoritmo *RangeSearch*, que parte del nodo raíz y recorre recursivamente todos los caminos que no pueden ser excluidos hasta los nodos hoja con objetos que satisfacen la consulta. Cuando el algoritmo accede a los objetos O_r del nodo N, la distancia del objeto padre O_p a la consulta se calcula una sola vez (las distancias de O_r a O_p están precalculadas en el índice) así es posible podar subárboles sin calcular más distancias.

3 TRABAJOS RELACIONADOS

El índice sobre espacios métricos M-tree fue mejorado en varias versiones posteriores con el fin de obtener mayor velocidad de indexado y recuperación. El *Slim-tree* [8] es una versión que acelera el indexado con el método de separación *minimal spanning tree*, y reduce la superposición de regiones con el algoritmo *Slim-down*, mejorando la performance de las consultas. El *M⁺-tree* [10] introduce el concepto de *dimensión clave*: particiona el espacio ocupado por un árbol en dos subespacios no superpuestos llamados *nodos gemelos*. En la búsqueda, la dimensión escogida como clave evita calcular distancias entre los objetos y por lo tanto, aumenta la efectividad de las podas. El *BM⁺-tree* [11] extiende el M⁺-tree mejorando el filtrado. En lugar de una dimensión clave utiliza dos para construir un *hiperplano binario* que particiona mejor los datos. El *Density-Based Metric tree* [9], o *DBM-tree*, minimiza la superposición en nodos de alta densidad, relajando el requerimiento de altura balanceada del árbol. La altura es mayor en las regiones más densas. Logra mayor desempeño en las búsquedas porque el árbol está ajustado a la distribución de los datos en las distintas regiones. El *DBM*-tree* [7] extiende al anterior, agregando en cada nodo una matriz con algunas distancias precalculadas entre los objetos que contiene con el fin de aumentar la velocidad de indexado y de búsqueda, porque permiten mayor número de podas de datos irrelevantes. En [5] se introduce los principios básicos y los resultados de una experimentación del *M²-tree*, un índice paginado y balanceado, que se construye sobre varios espacios métricos (atributos) a la vez y en una única estructura, permitiendo resolver eficientemente consultas que combinan arbitrariamente dichos atributos. El *XM-tree* propuesto aquí es un caso muy particular de este último índice, porque extiende un M-tree sobre espacios vectoriales, adaptando sus algoritmos a la búsqueda en la Web.

Con respecto a los motores de búsqueda en la web, su continuo crecimiento propone un desafío constante a la creación de índices veloces. Una solución actual es la utilización de índices invertidos, empleados por buscadores reconocidos como Google [2]. Sin embargo, estos índices presentan una desventaja, consecuencia de su estructura intrínseca: indexan de a una palabra. Cuando se resuelven consultas de varias palabras, primero se recuperan los documentos correspondientes a cada una de éstas, y luego éstos se someten a un proceso de unión o intersección. Para agilizar la velocidad de respuesta estos buscadores devuelven no sólo los documentos que satisfacen la estrategia de búsqueda, sino que devuelve también aquellos que posean alguno de los términos. El indexado con el XM-Tree propuesto en este trabajo mejora estos resultados.

4 PROPUESTA

El objetivo de este trabajo es utilizar las propiedades de los índices sobre espacios métricos para mejorar la calidad de los resultados de una búsqueda de información. El análisis sobre algoritmos de indexado sobre espacios métricos permitió vislumbrar la aptitud del M-tree para el entorno Web. Se optó entonces por extender el M-tree ya que al realizar una búsqueda, elige los recorridos analizando la consulta ingresada con información que contiene su estructura. Además, logra velocidad porque en este proceso va descartando los subárboles que no tienen datos próximos a la consulta. Los resultados obtenidos son correctos porque en el proceso de poda los subárboles rechazados contienen siempre datos irrelevantes, por definición del M-tree. Para alcanzar ambos

objetivos, velocidad y calidad de resultados, se elige una adecuada representación de las páginas Web y un conveniente criterio de similitud entre las mismas que posibiliten el empleo de un M-tree. A continuación se presentan las decisiones tomadas para esta propuesta.

Generación del Diccionario: El diccionario es un archivo de texto para medir las ocurrencias de las palabras en los documentos. Cada entrada corresponde a un término. Llamaremos *T_DIC* a la dimensión del vector diccionario.

Representación de los Documentos: Los documentos se representan como puntos en un espacio vectorial, cuya dimensión es el tamaño T_DIC del diccionario. Cada eje de este espacio representa una entrada del diccionario. Se utiliza el esquema TF para representar el documento[†]. El esquema TF es eficaz porque los documentos que comparten palabras muy frecuentes y del diccionario, quedan representados por vectores muy cercanos en cualquier distancia L_s. En dicha situación, un M-tree que cuenta como métrica una distancia L_s, agrupa estos documentos en el mismo subárbol y así acelera la resolución de las consultas conformadas por los términos compartidos.

Indexado: El índice utilizado es un M-tree cuya distancia de indexado es L_2, lo que le permite agrupar satisfactoriamente los documentos similares en el esquema TF. Se prefirió L_2 a las normas L_∞ y L_1 porque es más exacta en un espacio vectorial.

Búsquedas: Las consultas están conformadas por una o varias palabras, y se representan con un vector q de dimensión T_DIC con valores reales. Los términos que figuran en la entrada t del diccionario se indican con un valor no nulo en la componente t-ésima de q, el resto de las componentes se evalúan en 0. El principal objetivo del sistema es recuperar los documentos donde co-ocurran todas las palabras (o sus variantes morfológicas) ingresadas en la consulta. El algoritmo de búsqueda del M-tree obtiene los documentos cuyos vectores de frecuencias sean más próximos a q, según la norma L_2, por lo que hay que escoger valores adecuados para las componentes no nulas de q. Esto conlleva a calcular un q lo suficientemente cercano en el espacio euclídeo a todos los documentos que contengan los términos de la consulta y, de esta manera, permita recuperarlos en su totalidad. Como las frecuencias de mismas entradas pueden ser muy desproporcionadas entre sí en los documentos que las contienen, resulta imposible la obtención de un q óptimo porque habría que conocer de antemano los documentos que se pretenden recuperar. La situación se agrava si se pretende hacer una consulta por rango: ¿qué rango $r(Q)$ calcular? Ante estas circunstancias, se decidió utilizar un criterio de similitud entre los documentos y la consulta más relajado que L_2.

Para obtener los documentos que contengan todos los términos de la consulta, basta con chequear que las componentes que corresponden a dichos términos sean no nulas, las restantes componentes se pueden obviar. La resta entre 1 y la componente plasma este análisis: si el resultado es 1, ni la palabra ni sus variantes morfológicas figuran en el documento; si la diferencia es menor que 1 entonces el documento sí contiene al término. Entonces, el criterio de similitud es: asignar 1 a los elementos de q que corresponden a términos ingresados según el diccionario, calcular las restas de cada uno de dichos elementos y el correspondiente al vector del documento; si todas las restas dan menos que 1 entonces el documento satisface la consulta. Para extender este modelo al resto de las componentes fuera de la consulta y sin alterar los resultados, simplemente se les resta 0, o sea la componente de q. Como las frecuencias por definición varían entre 0 y 1 y entre todas suman 1, estas diferencias no alcanzarían al valor 1 (salvo un caso extremo) y no afectarían el criterio.

El nuevo criterio de similitud entre documentos y consultas se puede formular con el cálculo de la norma L_∞ entre el vector del documento y el vector q de la consulta. Si el resultado es menor que 1 entonces el documento contiene todos los términos de la consulta, en caso contrario, se descarta. Como el algoritmo de búsqueda del M-tree por definición utiliza la misma distancia con que se construyó el índice, para usar el criterio propuesto habría que armar el árbol empleando la norma L_∞

† Se descartó usar el esquema TF/IDF porque requiere para el cálculo de cada componente el número de ocurrencias del término en el resto de los documentos. Así, al ingresar un nuevo documento, habría que recalcular los vectores de todos los documentos, por lo que el índice no sería dinámico.

como métrica. Esta opción no resulta atractiva porque determina la proximidad de dos documentos teniendo en cuenta la frecuencia de un solo término, aquél cuya diferencia de frecuencias en ambos documentos es máxima, pudiendo llegar a considerar cercanos documentos que no comparten términos. La otra alternativa es seguir empleando como distancia de indexado a la norma L_2 pero modificar el algoritmo de búsqueda para que compare los documentos y la consulta con L_∞. Este cambio se traslada al criterio de poda que no podrá seguir usando L_2, si no que requerirá cierta información adicional para dirimir con una distancia distinta a la de indexado. Dicha información adicional concierne a las *componentes* de los datos del subárbol tratadas por separado, por lo que debe estar en el nodo interno raíz del subárbol analizado. Los cambios mencionados en el algoritmo de búsqueda y en la estructura de los nodos constituyen la propuesta de extensión del M-tree.

4.1 Propuesta de M-tree Extendido: XM-tree

El *XM-tree* es un índice sobre espacios vectoriales para la búsqueda en la web. Es una extensión del M-tree con la particularidad que trata las componentes de los vectores indexados por separado. Las extensiones sobre el M-tree original comprenden información adicional en los nodos y las correspondientes modificaciones en los algoritmos de construcción para mantenerla. La información que se añade hace referencia a las distancias entre las componentes de los datos indexados tratadas separadamente, con el fin de preservar estas medidas y emplearlas para posibles podas en el algoritmo de búsqueda. El objetivo de esta propuesta es lograr un índice sobre páginas web que utilice la norma L_2 como distancia de indexado y la norma L_∞ como distancia de consulta.

Se agrega al nodo interno el *vector de radios de cobertura* $rv(O_r)$. Si n es la dimensión de los datos, O_r y O_j tienen componentes O_{ri} y O_{ji} respectivamente, y consideramos una distancia f sobre el espacio de las componentes, debe valer para cada componente que $rv_i(O_r)$: $f(O_{ri}, O_{ji}) \leq rv_i(O_r)$ $i = 1, 2, ..., n$ $\forall O_j \in T(O_r)$. La distancia entre componentes f es una norma L_s, que, como se trata de una sola dimensión, para cualquier s coincide con el valor absoluto de la diferencia. Además se agrega el *vector de distancias* $dv(O_r, P(O_r))$, con las distancias entre las componentes del objeto ruteador O_r y las del padre $P(O_r)$. Cada componente vale: $dv_i(O_r,P(O_r)) = f(O_{ri},P(O_r)_i)$, $i = 1,2, ..., n$.

Entonces *cada nodo interno* del XM-tree queda conformado de la siguiente manera:

O_r	objeto ruteador
$ptr(T(O_r))$	puntero a la raíz de $T(O_r)$
$r(O_r)$	radio de cobertura de O_r
$rv(O_r)$	*vector con radios de cobertura de las componentes de O_r*
$d(O_r, P(O_r))$	distancia de O_r a su padre
$dv(O_r, P(O_r))$	*vector con distancias entre componentes de O_r a su padre*

Los nodos hoja no sufren modificaciones respecto al M-tree. Las entradas son: O_j*:* objeto de la base de datos, $oid(O_j)$: identificador del objeto y $d(O_j, P(O_j))$: distancia de O_j a su padre. Al igual que el M-tree, el objetivo de los algoritmos de búsqueda de esta extensión es reducir el número de nodos accedidos y de distancias calculadas, gracias a las distancias precalculadas almacenadas en $rv(O_r)$ y $dv(O_r,P(O_r))$ de los nodos internos y a la desigualdad triangular.

El XM-tree opera sobre datos de un espacio vectorial de dimensión n, con una distancia f sobre el espacio de las componentes. Dados $Q = (Q_1, Q_2, ..., Q_n)$ a buscar en el espacio vectorial y un vector $rv(Q)$ de radios de búsqueda con componentes $rv_i(Q)$, la consulta estricta $(Q, rv(Q))_f$ selecciona todos los objetos O_j de la base de datos tal que $f(O_{ji}, Q_i) < rv_i(Q)$, $\forall i = 1, 2, ..., n$.

El algoritmo modificado *RangeSearch* parte del nodo raíz y recorre todos los subárboles no excluidos por el criterio de poda hasta los nodos hoja con objetos que satisfacen la consulta.

En la línea 3 el algoritmo accede a los objetos O_r del nodo N. Las distancias entre las componentes Q_i y las del objeto padre O_{pi} se calculan una sola vez para i de 1 a n, y las distancias entre O_{ri} a O_{pi} figuran precalculadas en el vector $dv(O_r, O_p)$. Así, se pueden podar subárboles sin

calcular más distancias. La poda se efectúa en la línea 6 si se da la condición: si $f(O_{ri}, Q_i) \geq rv_i(Q) + rv_i(O_r)$ entonces, para cada O_j en $T(O_r)$ vale $f(O_{ji}, Q) \geq rv_i(Q)$. De esta manera $T(O_r)$ puede ser descartado. En la línea 9 se chequean los nodos hoja con objetos O_j. Aquí también las distancias entre las componentes O_{pi} y Q_i se calculan una vez y los elementos $dv_i(O_j, O_p)$ figuran en la estructura. La búsqueda en subárboles irrelevantes se evita en las líneas 4 y 10 empleando el siguiente resultado, considerando que los radios de cobertura $rv_i(O_j)$ de los datos es 0: Si $|f(O_{pi}, Q_{ii}) - dv_i(O_r, O_p)| \geq rv_i(Q) + rv_i(O_r)$ entonces $f(O_{ri}, Q_{ii}) \geq rv_i(Q) + rv_i(O_r)$. Estas dos condiciones son válidas porque son casos particulares de lemas propuestos por [4].

Algoritmo RangeSearch(node N, query_object Q, search_radius rv(Q))

```
1.  Let Op be the parent object of node N
2.  If N is not a leaf then
3.    For each object Or in N do
4.      If (|f(Opi, Qi) – dvi(Or, Op)| < rvi(Q) + rvi(Or) i=1..n) then
5.        Compute f(Ori, Qi) i=1..n
6.        If (f(Ori, Qi) < rvi(Q) + rvi(Or) i=1..n) then
7.          RangeSearch(*ptr(T(Or)), Q, rv(Q))
8.  else   // N is a leaf
9.    For each node child Oj in N do
10.     If (|f(Opi, Qi) – dvi(Oj, Op)| < rvi(Q) i=1..n) then
11.       Compute f(Oji, Qi) i=1..n
12.       If (f(Oj, Qi) < rvi(Q) i=1..n) then add oid(Oj) to the result set
13. End
```

Los algoritmos de construcción del XM-tree, *Insert* y *Split*, son los del M-tree original, con el agregado de sentencias para el mantenimiento de los nuevos datos $rv(O_r)$ y $dv(O_r, P(O_r))$.

Para llevar a cabo las búsquedas, una vez representadas las páginas web con vectores del esquema TF, se construye el XM-tree empleando como distancia de indexado d la norma L_2 y como distancia entre componentes f una norma L_s, que, como se trata de una sola dimensión, para cualquier s coincide con el valor absoluto de la diferencia. El algoritmo de búsqueda se invoca pasando un vector de radios con componentes $rv_i(Q)$ de valor 1. Con esta táctica se logra el objetivo de indexar los datos con L_2 y buscar con L_∞, esto último a partir de la siguiente equivalencia entre el criterio de búsqueda propuesto y la consulta por rango estricta $(Q, rv(Q))_f$ del XM-tree: $L_\infty(Q, O) < 1 \leftrightarrow \max\{|Q_i - O_i|\} < 1 \leftrightarrow |Q_i - O_i| < 1 \leftrightarrow f(Q_i, O_i) < 1 \leftrightarrow f(Q_i, O_i) < rv_i(Q) \quad \forall i=1, 2, ..., n$ ‡.

5 EVALUACIÓN EXPERIMENTAL

5.1 Implementación

El sistema propuesto en este trabajo ha sido implementado en C^{++} utilizando DJGPP [6]. Se construyó un diccionario de 1000 entradas, y considerando las variantes morfológicas de cada entrada, el diccionario agrupa unos 7000 términos. Se calculan los vectores TF de las páginas web a indexar, considerando solamente el texto visible de las páginas. Se genera un archivo de salida con los siguientes datos para cada página: número identificador, vector de frecuencias, título, URL y ubicación del respaldo. Este prototipo cuenta con procedimientos que modelan los algoritmos de inserción, separación y búsqueda del XM-tree. Al iniciarse lee el archivo de datos generado, construye el árbol dinámicamente y solicita el ingreso de consultas vía línea de comandos. La aridad M del árbol es un valor fijo del programa. La implementación ofrece al usuario la posibilidad de utilizar operadores lógicos. El operador por defecto es 'AND'.

En la implementación del algoritmo de separación se escogieron las políticas m_RAD e

‡ El logro de esta equivalencia justifica la decisión de emplear la consulta por rango estricta, ya que la consulta por rango original no sería útil porque permitiría alcanzar el valor de los radios $r_{vi}(Q)$.

Hiperplano Generalizado para modelar los algoritmos de promoción y partición respectivamente. En m_RAD, el algoritmo "mínima suma de radios" promueve los objetos cuya suma de radios de cobertura $r(Op_1) + r(Op_2)$ es mínima. Hiperplano Generalizado asigna cada objeto $O_j \in S$ al objeto promocionado más próximo. Esta estrategia es no balanceada. Por el momento se han considerado sólo estas dos políticas, pero como trabajo futuro se podrían evaluar las otras políticas.

Una vez procesada la consulta, el prototipo presenta datos de los tres tipos de búsquedas: la búsqueda en el XM-tree; una búsqueda exhaustiva, con la consulta original sobre todas las páginas indexadas; y una búsqueda que emula un sistema de índices invertidos. En la interfaz se presenta información sobre la estructura del XM-tree construido: cantidad de nodos internos, de nodos hoja y de datos indexados. Para analizar la eficiencia de la búsqueda en el árbol se muestran: cantidad y porcentaje de nodos recorridos y de nodos podados. Para verificar la efectividad de la búsqueda en el XM-tree se muestran la cantidad de resultados, la Precisión y el Recall. Luego del proceso de filtrado, se muestran nuevamente cantidad de resultados, Precisión y Recall, los cuales serían los valores finales alcanzados por el sistema. La estrategia de búsqueda con la que se invocó al algoritmo, se compara con los vectores de todas las páginas usando el criterio de similitud L_∞. Aquellas páginas que verifican el criterio y no se retornaron en la búsqueda, o que no lo verifican y sí fueron retornadas, son falencias del XM-tree y su cantidad se muestra con el título 'errores'.

La búsqueda exhaustiva corresponde a la aplicación de la consulta ingresada al texto de cada página. El objetivo es mostrar la efectividad de las siguientes propuestas hechas en este trabajo: el esquema TF como mecanismo de representación de páginas, el criterio de similitud L_∞ entre página y consulta, y el XM-tree como método de indexado. La cantidad de resultados diferentes entre esta búsqueda y la principal se muestran en pantalla. El número de resultados aquí obtenidos se considera el número total de documentos relevantes en la colección y se emplea para calcular los indicadores Precisión y Recall. La última búsqueda encuentra los datos que recuperaría un sistema con índices invertidos. La diferencia entre los datos recuperados por la primera instancia de esta búsqueda y la búsqueda principal también se muestra en la interfaz. La intención de esta comparación es verificar la mayor eficiencia del XM-tree sobre los índices invertidos. Los datos se ordenan de acuerdo al máximo producto interno con los vectores q, de mayor a menor.

5.2 Experimentación

Para la experimentación se realizaron consultas en el prototipo, con el fin de mostrar la performance del sistema en un ambiente similar al entorno web real. Las páginas que conforman el cuerpo de documentos que indexa el XM-tree en este testing constituyen un conjunto muy pequeño en comparación con el total que compone la web indexada, pues su objetivo es solamente experimental. Por esta misma razón, se trató de construirlo de la manera más representativa posible.

Para simular de la mejor manera alcanzable la heterogeneidad temática que caracteriza a Internet en una muestra reducida, se recurrió a la recolección de páginas web reales que traten varios temas distintos pero que compartan algunos términos significativos. Los documentos que componen la muestra se obtuvieron mediante consultas formuladas a Google. El cuerpo de documentos construido contiene poco más de 200 páginas, es suficientemente representativo porque se obtuvieron de la web real, la gran mayoría concierne a 8 tópicos de bastante interés, y a su vez, algunos de estos temas pueden estar relacionados entre sí por compartir términos significativos.

Un usuario que pretende recuperar información sobre el cáncer de pulmón podría hacerlo a través de la consulta: (cáncer OR tumor) AND (pulmón OR pulmones OR pulmonar). Una vez ejecutadas las búsquedas la información mostrada es la siguiente:

```
=== BUSQUEDA EN EL SISTEMA BASADO EN XM-TREE ============================
estructura        : 8 ints, 47 hojas, 224 datos, total 279
revisados         : 7 ints (87.50 %) 14 hojas (29.79 %) 96 datos (42.86 %)
podados           : 30                errores   : 0
Resultados XM-tree : 29               Precision : 1.00          Recall : 1.00
```

```
=== COMPARACION CON BUSQUEDA CON INDICE INVERTIDO (173 resultados) ==========
no recuperados por el XM-tree          : 144
no recuperados por el indice invertido  : 0
=== COMPARACION CON BUSQUEDA EXHAUSTIVA (29 resultados) ====================
no recuperados por el XM-tree          : 0
no recuperados por la busqueda exhaustiva : 0
```

En el campo *estructura* se informa la cantidad de nodos internos, nodos hoja y datos del árbol construido, la cantidad de datos coincide con el total de documentos indexados, en este caso 224. En la siguiente línea se muestran la cantidad y porcentaje de nodos *revisados* de cada tipo en la búsqueda XM-tree. La revisión de un nodo implica el cálculo de normas *f* del XM-tree. Un bajo porcentaje de nodos revisados indica un buen agrupamiento de los datos similares en las hojas del XM-tree, por lo que el algoritmo de búsqueda encamina la consulta hacia pocas hojas y poda muchos nodos. La cantidad de nodos *podados* corresponde al número de subárboles descartados por el algoritmo RangeSearch siguiendo los criterios de poda, o sea, nodos internos y hojas sin datos relevantes. En este caso es bastante alto, 30 podas sobre 55 nodos (internos y hojas), por la alta especificidad de la consulta en la colección. El número *errores* es nulo porque la búsqueda en el árbol recuperó correctamente todos los datos que verifican el criterio de similitud L_∞ con la consulta. Los indicadores *Precisión* y *Recall* de la recuperación XM-tree en sus valores óptimos reflejan la alta efectividad de la búsqueda en dicho índice: obtuvo todos y sólo los documentos relevantes, como se verifica en la comparación con la búsqueda exhaustiva.

La búsqueda que emula un sistema de índices invertidos obtiene 173 datos correspondientes a las posting lists de los términos ingresados, los cuales posteriormente se someterían al proceso de mezclado e intersección. Este valor se compara con el número de resultados XM-tree, en este caso 29, que como es mucho menor verifica el mejor desempeño del XM-tree para la consulta ingresada.

El archivo de páginas se reordenó aleatoriamente, para analizar la dependencia o no de la perfomance del XM-tree respecto al orden de inserción de los datos. Para esto, se construyeron cinco conjuntos de datos con distintas ordenaciones de las páginas que lo conforman. Se obtuvieron los mismos valores de Precisión y Recall, para cada uno de los conjuntos de datos, por lo que no es importante el orden de los documentos de entrada. Posteriormente se procedió a analizar la eficiencia de la búsqueda en el árbol construido sobre cada conjunto.

En la Figura 1 cada línea corresponde a una de las consultas e indica el porcentaje de datos revisados en cada conjunto. Los conjuntos de datos son numerados de 1 a 5. Como se aprecia, una consulta no registra diferencias abruptas de porcentaje entre un conjunto y otro. Esto demuestra, acerca del grado de eficacia del indexado, que el árbol agrupa de la igual manera páginas similares sin importar el orden del conjunto sobre el que se construyó. Por lo tanto, los valores similares de Precisión, Recall y datos revisados permiten concluir que la eficiencia y eficacia del sistema es independiente del orden en que se indexan las páginas. El segundo aspecto analizado es la aridad *M* del árbol, o sea, el tamaño de cada nodo. En la Figura 2, se muestran los porcentajes de datos revisados en cada consulta para diferentes aridades consideradas: 5, 10, 15, 20 y 25.

En la Figura 3 se presentan las cantidades totales de nodos de distintos árboles con los valores mencionados de *M*. Se observa cómo a mayor *M*, el árbol ocupa menos espacio pero revisa más datos, porque las hojas contienen muchos datos y el algoritmo de búsqueda las chequea por completo. Una pérdida de eficiencia aún más importante es la menor capacidad de agrupar datos de un XM-tree con alta aridad, porque el agrupamiento de datos similares en subárboles distintos sucede luego de que un nodo se desborda durante la inserción de uno nuevo, lo que no se da a menudo en un árbol con nodos de gran tamaño. Así se concluye que, si bien la elección de *M* no afecta la calidad de los resultados se debe escoger, en principio, un valor intermedio de acuerdo al tamaño de la colección (aquél cuyo árbol ocupe una cantidad de espacio razonable). Con la sucesiva incorporación de nuevos datos a la colección y su inserción en el árbol, el aprovechamiento razonable de espacio mermará, pero no se perderá eficiencia en la resolución de consultas.

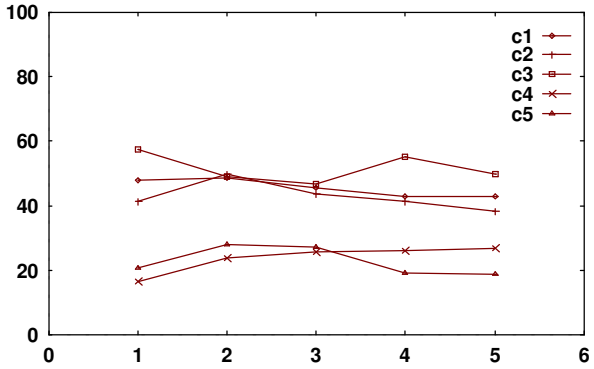


Figura 1. Porcentaje de datos revisados por cada conjunto de datos.

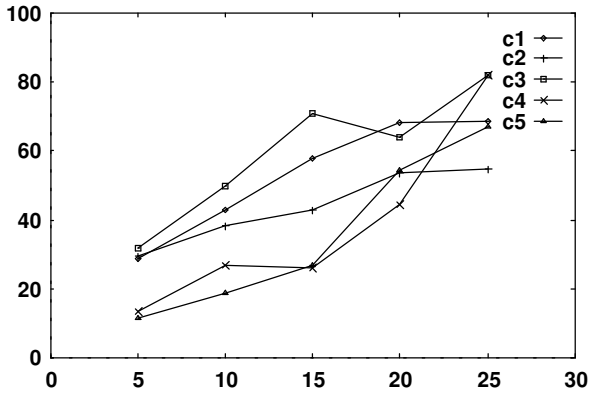


Figura 2. Porcentajes datos revisados por *M*.

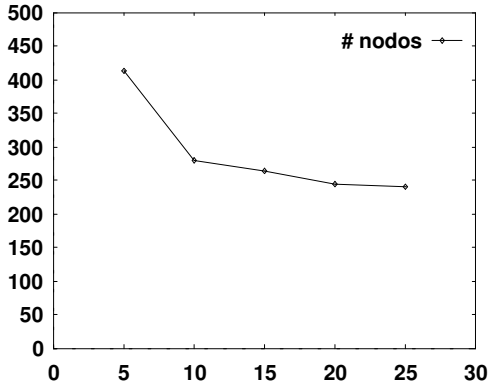


Figura 3. Número de nodos por aridad *M*

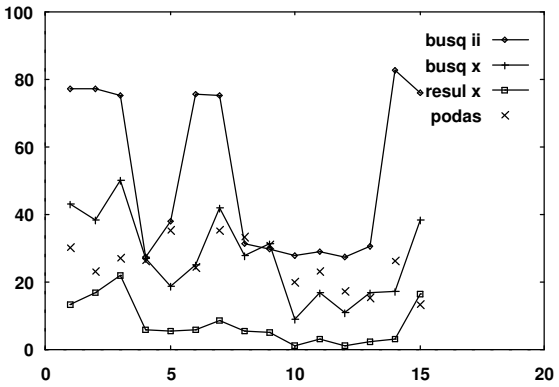


Figura 4. Porcentajes de datos revisados, resultados y podas por consulta

Para la siguiente etapa experimental se preservó el XM-tree de aridad 10, debido a que con un valor menor el desaprovechamiento de espacio ya es muy ostensible, como muestra la Figura 3 en $M = 5$. Con valores más altos, la revisión de datos no relevantes tiende a aumentar (Figura 2).

La última fase de la experimentación consistió en comparar la performance de este sistema y un sistema de índices invertidos. A las cinco consultas utilizadas para las primeras experiencias, se agregaron diez consultas que atañen al resto de los principales temas de la colección, incluso algunas muy específicas dentro de un mismo tema. En la Figura 4 las consultas están numeradas de 1 a 15 en las abscisas, para cada una se muestran los siguientes valores en las ordenadas: el porcentaje de datos recuperados por un sistema de índices invertidos; el porcentaje de datos revisados en la búsqueda del XM-tree; el porcentaje de datos recuperados del XM-tree; y el número de nodos podados por la búsqueda en el árbol. En dicha gráfica se observa cómo la cantidad de datos revisados por la búsqueda XM-tree es proporcional a la cantidad de resultados relevantes en cada consulta. Esto se debe al agrupamiento de documentos similares en las hojas del XM-tree, sólo se revisan las hojas con documentos próximos a la consulta. Esto no ocurre en los índices invertidos, donde la cantidad de datos revisados depende directamente del término ingresado con más ocurrencias en la colección, y que, por tal motivo, contiene posting lists de gran tamaño. En cuanto a la revisión de datos de la colección durante una búsqueda, el sistema propuesto realiza un gasto proporcional a la cantidad de documentos relevantes a la consulta.

6 CONCLUSIONES

Velocidad en la recuperación y calidad de los resultados son dos propiedades necesarias en cualquier sistema de Recuperación de Información. Los espacios métricos cuentan con índices que permiten la recuperación de objetos cercanos a uno dado de una forma rápida y bastante exacta, por

lo que resultan estructuras prometedoras sobre las cuales se pueden construir motores de búsqueda.

En este trabajo se propuso el XM-tree que es un índice sobre espacios vectoriales. Es una extensión del M-tree, y al igual que éste, es una estructura paginada, balanceada y dinámica que indexa datos de un espacio métrico (en particular, un espacio vectorial), resuelve consultas por rango dado un objeto de consulta, la ejecución de la búsqueda es optimizada de forma que reduce el número de datos leídos y de distancias calculadas, y es apto para datos de espacios vectoriales de alta dimensión. La extensión consiste en el tratamiento por separado de las componentes de los vectores indexados, y tiene como objetivo adaptar el algoritmo de búsqueda del M-tree a un criterio de similitud de la Recuperación de Información en la Web. Esta estructura indexa documentos Web representados en el esquema TF, emplea como distancia de indexado la norma L_2 y resuelve las búsquedas aplicando como criterio de similitud entre consulta y documento la norma L_∞.

El XM-tree logra un alto rendimiento en cuanto a calidad de resultados en un proceso de Recuperación de Información en la Web, alcanzando buenos valores de Precisión y Recall. Además, la eficiencia de las búsquedas ofrece importantes mejoras sobre los espacios vectoriales e índices invertidos. Respecto a los espacios vectoriales, el XM-tree agrupa adecuadamente los documentos permitiendo recorrer sólo los próximos a la consulta. A diferencia de los índices invertidos, revisa una fracción de la colección proporcional al conjunto de documentos relevantes. Los resultados experimentales confirmaron la concreción de ambas metas: calidad en los resultados y velocidad en la resolución de consultas.

Una extensión futura es el borrado real de datos, ya que actualmente las páginas fuera de línea siguen presentes en el índice y figuran como accesibles desde un caché, tal cual lo hace el M-tree.

REFERENCIAS

[1] R. Baeza-Yates y B. Ribeiro-Neto. (eds.). Modern Information Retrieval. *ACM Press, New York*, 1999.

[2] S. Brin y L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *In Proc. of the Seventh International World Wide Web Conference, vol. 30 of Computer Networks and ISDN Systems*, 107–117, 1998.

[3] E. Chávez, G. Navarro, R. Baeza-Yates y J. L. Marroquín. Searching in Metric Spaces. *ACM Computing Surveys*, 33(3):273–321. 2001.

[4] P. Ciaccia, M. Patella y P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Proc. of the 23rd Conference on Very Large Databases (VLDB'97)*, 426–435, 1997.

[5] P. Ciaccia y M. Patella. The M^2-tree: Processing Complex Multi-Feature Queries with just one Index. URL www.ercim.org/publication/ws-proceedings/DelNoe01/16_Ciaccia.pdf, 2000.

[6] D. J. Delorie. DJGPP. Copyright (c) 2003 URL http://www.delorie.com/djgpp/

[7] A. Ocsa y E. Cuadros-Vargas. DBM*-Tree: An Efficient Metric Acces Method. *In Proc. of ACM Southeast Regional Conference*, 401–406, 2007.

[8] C. Traina Jr., A. Traina, B. Seeger y C. Faloutsos. Slim-trees: High Performance Metric Trees Minimizing Overlap between Nodes. *In Proc. of 7th International Conference on Extending Database Technology*, *LNCS*, 1777, 51–68, 2000.

[9] M. R. Vieira, C. Traina Jr., F. J. T. Chino y A. J. M. Traina. DBM-Tree: Trading Height-Balancing for Performance in Metric Access Methods. *Journal of the Brazilian Computer Society*, v. 11, n. 3, p. 20 p, 2006.

[10] X. Zhou, G. Wang, J. Xu Yu y G. Yu. M^+-tree: A New Dynamical Multidimensional Index for Metric Spaces. *In Proc. of the 14th Australasian Database Conference*,161 – 168, 2003.

[11] X. Zhou, G. Wang, X. Zhou y G. Yu. BM^+-tree: A Hyperplane-based Index Method for High-Dimensional Metric Spaces. *In Proc. of 10th International Conference Database Systems for Advanced Applications*, LNCS, 3453, 398–409, 2005.

Aplicaciones veterinarias del procesamiento de imágenes

Mariana del Fresno[1] y Alejandro Clausse[1,2]
[1]Universidad Nacional del Centro, Tandil, Argentina
[2]CNEA-CONICET
{mdelfres, clausse}@exa.unicen.edu.ar

Resumen

Se presenta el desarrollo de un algoritmo de procesamiento de imágenes para el soporte de detección de celo vacuno. El método de detección se basa en el análisis del deterioro de pintadas lumbares, mediante la segmentación de fotografías color tomadas a la salida de un tambo. Inicialmente, las regiones de interés son aisladas del resto de la imagen en función de valores de umbral definidos en los espacios de color YUV e YIQ, generando un conjunto de componentes conectadas. Luego, se calcula un conjunto de características para posibilitar una evaluación cuantitativa de los objetos segmentados. Finalmente, la imagen es clasificada mediante una regla de decisión basada en el análisis de las diferencias entre las medidas obtenidas a partir de la imagen y las provenientes de un conjunto de imágenes manualmente segmentadas, de acuerdo a la evaluación de expertos. Este enfoque constituye una valiosa alternativa para la mejora del control de celo, ya que la ardua tarea de observación visual de los animales en el tambo puede ser reemplazada por el análisis automático de imágenes capturadas en ambientes controlados. Los resultados experimentales a partir de las segmentaciones obtenidas con este método resultan altamente satisfactorios, permitiendo una clasificación precisa de las imágenes con un bajo costo computacional.

Palabras clave: Procesamiento de señales, segmentación de imágenes, clasificación.

Abstract

This paper presents the development of an algorithm of image processing which was applied to a decision support tool for estrus detection in cattle. The detection method is based on the analysis of rubbed off lumbar paintings, by means of the segmentation of color pictures taken to cows in a cattle ranch. Firstly, the regions of interest are isolated from the rest of the image based on threshold functions defined in the YUV and YIQ color spaces, producing a set of connected components. Then, a set of features is computed to enable a quantitative evaluation of the segmented objects. Finally, the image is classified by means of a decision rule based on the analysis of the differences between the computed measures and a set of ideally segmented images, according to experts' assessment. This approach constitutes a valuable alternative to improve this process, as it may replace the visual observation by the automatic analysis of pictures taken to cows in controlled environments. Experimental results show that the segmentations obtained with this method are highly satisfactory and they allow a precise classification of the images with low computational complexity.

Keywords: Signal processing, image segmentation, classification.

1. INTRODUCCIÓN

La segmentación es la operación de identificación y separación de objetos significativos contenidos en una imagen. Esta es una etapa sumamente importante en la automatización de procesos de reconocimiento y determina el eventual éxito o fracaso de tareas de más alto nivel como clasificación, análisis y visualización [2].

Se han propuesto diferentes técnicas de segmentación [6, 9]. El umbralado es uno de los métodos más simples y más comúnmente utilizados [10]. Se basa en una clasificación estadística de los niveles de gris de la imagen, suponiendo que todos los puntos con valores de intensidad dentro de un cierto rango corresponden al mismo objeto. Por otro lado, los enfoques basados en bordes buscan la detección de discontinuidades de los niveles de gris que idealmente se asocian a los límites entre diferentes componentes [2]. Otro tipo de enfoques se basa en la determinación de regiones homogéneas dentro de la imagen, por medio de la incorporación de aquellos píxeles que satisfacen cierto criterio de conectividad y similitud [1]. Finalmente, existen varios métodos que combinan dos o más de los enfoques anteriores y que resultan más complejos, como por ejemplo los basados en *snakes* [7] o redes neuronales [9]. Por lo general, la mayoría de estos algoritmos resultan computacionalmente intensivos y por lo tanto poco convenientes para aplicaciones en tiempo real.

Recientemente se ha manifestado un interés creciente por su aplicación a imágenes color, debido a que éstas generalmente aportan mayor contenido de información, facilitando la identificación de objetos [11]. Entre los algoritmos desarrollados, se encuentran algunas técnicas ad hoc que utilizan conocimiento específico sobre la naturaleza de la información de color. Por otro lado, también pueden explorarse extensiones convenientes de las técnicas usadas para imágenes en tonos de gris, aplicando por ejemplo el algoritmo a cada componente del espacio de color (*RGB*, *YIQ*, etc.) para luego combinar los resultados individuales.

El campo de aplicación de la segmentación de imágenes color es sumamente amplio y se han propuesto interesantes usos en diversas áreas. En particular, la detección de celo en bovinos en base a pintadas lumbares constituye un ejemplo práctico de estas aplicaciones. La técnica consiste en pintar una banda de 20 cm de largo por 5 cm de ancho a lo largo de la base de la cola del animal. Durante la monta, la pintura se desgasta y se pierde gradualmente. De esta manera, la ausencia o deterioro de la pintura en la zona lumbar puede ser usada para determinar un estado de celo [8]. La detección se realiza generalmente mediante la observación visual de operadores entrenados ya que no existen dispositivos automáticos para este propósito. Al disponer de una herramienta de segmentación de imágenes color provenientes de las pinturas lumbares en tiempo real, se podrían implementar sistemas automáticos de detección de celo que permitan reducir costos.

En este trabajo se presenta un algoritmo computacionalmente eficiente para segmentación y clasificación de imágenes color. El método ha sido aplicado a la segmentación de pinturas sobre el área lumbar de vacunos, seguida por la extracción y comparación de algunas características de las imágenes a fin de extraer resultados significativos. Esta herramienta provee un sistema automático novedoso para el soporte de decisiones capaz de determinar en tiempo real el estado de celo de los animales, lo que permite mejorar el diagnóstico y obtener mayores ganancias a los productores.

El artículo se organiza de la siguiente manera. En la Sección 2, se discute el algoritmo de segmentación basado en umbralado a partir de transformaciones no lineales de los espacios de color. Las etapas de extracción y clasificación se describen en la Sección 3. En la Sección 4, se analiza detenidamente la performance del método usando diferentes grupos de imágenes. Finalmente, en la Sección 5 se presentan las conclusiones generales del trabajo.

2. SEGMENTACIÓN DE IMÁGENES COLOR

A pesar de la disponibilidad de otras opciones de segmentación, el umbralado se encuentra entre las técnicas más populares para segmentar imágenes debido a su baja complejidad computacional y buena performance. El umbralado frecuentemente provee un enfoque simple y apropiado para la detección de objetos del fondo usando el histograma de la imagen. Si el rango de intensidades se encuentra bien diferenciado, esta tarea puede realizarse efectivamente eligiendo un valor de umbral adecuado en el valle entre las dos modas dominantes. El problema es más complicado cuando las distribuciones se superponen y por lo tanto no es simple determinar el valle [5].

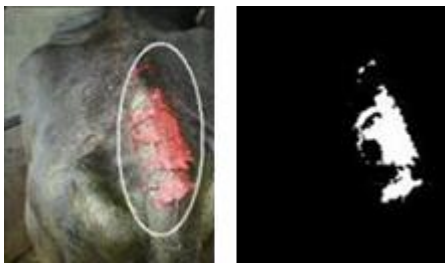
En su forma más simple, cada píxel de una imagen *f(i,j)* es comparado con un umbral *T*, y luego es clasificado para obtener una imagen binaria *I(i,j)* definida como:

$$I(i,\ j) = \begin{cases} 1 & \text{if } f(i,j) > T \\ 0 & \text{if } f(i,j) \leq T \end{cases} \quad (1)$$

esto es, los píxeles rotulados con 1 corresponden a objetos, mientras que los rotulados con 0 corresponden al fondo.

Si bien el umbralado es una técnica fácil de aplicar a imágenes en tonos de gris, pueden aparecer algunas complicaciones en el caso de imágenes color. El color se representa comúnmente por una tripla de componentes escales rojo, verde y azul (*R*, *G*, *B*) en un espacio ortogonal, o alguna transformación de ellos. Detectar clusters de puntos en este espacio involucra el análisis de picos y valles de tres histogramas diferentes o determinar un umbral apropiado en un histograma tridimensional, el cual puede requerir un alto costo computacional [3]. Por otra parte, el espacio *RGB* es apropiado para visualizar color, pero no es conveniente para muchas tareas de procesamiento de imágenes a causa de la alta correlación entre sus componentes. Contrariamente, el espacio *YIQ* tiene en cuenta la mayor sensibilidad humana a los cambios en luminosidad (*Y*) que en la información sobre el color (*I* y *Q*). Estas componentes se encuentran desacopladas, de manera que pueden ser procesadas independientemente. El espacio *YUV* es una versión levemente diferente del YIQ pero con las mismas ventajas [4].

Aunque el algoritmo desarrollado puede ser aplicado a diferentes áreas con pocas modificaciones, el presente estudio se enfoca en imágenes con zonas artificialmente pintadas de color rojo, amarillo, verde o azul (de acuerdo a los colores utilizados por los veterinarios en los diferentes ciclos de celo) sobre fondos en tonos marrón, negro y blanco (correspondientes a los posibles cueros vacunos). El principal objetivo es separar automáticamente las regiones pintadas del fondo para permitir su evaluación posterior (Figura 1). Inicialmente, las variables de crominancia (*u,v* en el espacio *YUV* e *i,q* en el espacio *YIQ*) fueron analizadas para todos los colores posibles de pintura y cuero, tratando de encontrar valores de umbral apropiado. Sin embargo, ni *YUV* ni *YIQ* poseen un rango de valores que puedan ser usados para discriminar efectivamente entre objetos y fondo. Por esta razón, se estudiaron varias funciones de color de las variables de crominancia a fin de identificar un campo escalar adecuado que permita asegurar una correcta segmentación de la imagen.



(a) (b)

Figura 1. Segmentación de una imagen típica: banda de pintura sobre la base de la cola de un animal (a), objeto de interés segmentado del fondo (b)

A fin de evaluar los resultados de la segmentación de diferentes funciones de color, se generó un conjunto de 1000 imágenes sintéticas semejando escenas libres de error. Este análisis permitió analizar el comportamiento de las funciones que lograron distinguir correctamente los diferentes colores utilizados para las pinturas y los asociados con los distintos cueros de los animales. Como ejemplo, la Figura 2 muestra cuatro histogramas correspondientes a la función $\sqrt{u^2+v^2}$, mostrando una clara distinción entre los dos grupos.

Luego de haber procesado las imágenes con las distintas funciones propuestas, se eligió un subconjunto de ellas según su performance. Las funciones seleccionadas fueron: $k \max(|u|,|v|)$ y $\sqrt{u^2+v^2}$, en el espacio *YUV*, y $k_1 \max(|i|,|q|)+k_2 \min(|i|,|q|)$ y $\sqrt{i^2+q^2}$ en el espacio *YIQ*, ya que con ellas se logró segmentar aproximadamente el 97% de las imágenes de prueba, para los cuatro colores de pintura y con un rango de valores bastante amplio. Los valores de k, k_1 y k_2 fueron obtenidos mediante ensayos numéricos, verificando que con k= 11 para el primer caso, y k_1= 10, k_2= 2 para el segundo se logran segmentaciones exitosas con una alta tasa de eficiencia. Este conjunto de funciones fue utilizado en pruebas con imágenes experimentales, como será explicado en la Sección 4.

3. PROCESAMIENTO POST-SEGMENTACIÓN

La imagen binaria obtenida en la etapa de segmentación se procesa mediante una operación de erosión para eliminar los posibles puntos espurios que pueden surgir durante el umbralado [2]. Esta operación morfológica puede reducir levemente el tamaño de los objetos, pero a la vez contribuye a revelar su forma básica, lo cual se requiere en la etapa de clasificación. Luego de este proceso, las componentes conectadas que permanecen son rotuladas a fin de poder determinar las regiones de interés [5]. Entonces, estas regiones son evaluadas mediante sus características geométricas y se las compara contra medidas ideales (Figura 3), como se describe en las siguientes subsecciones.

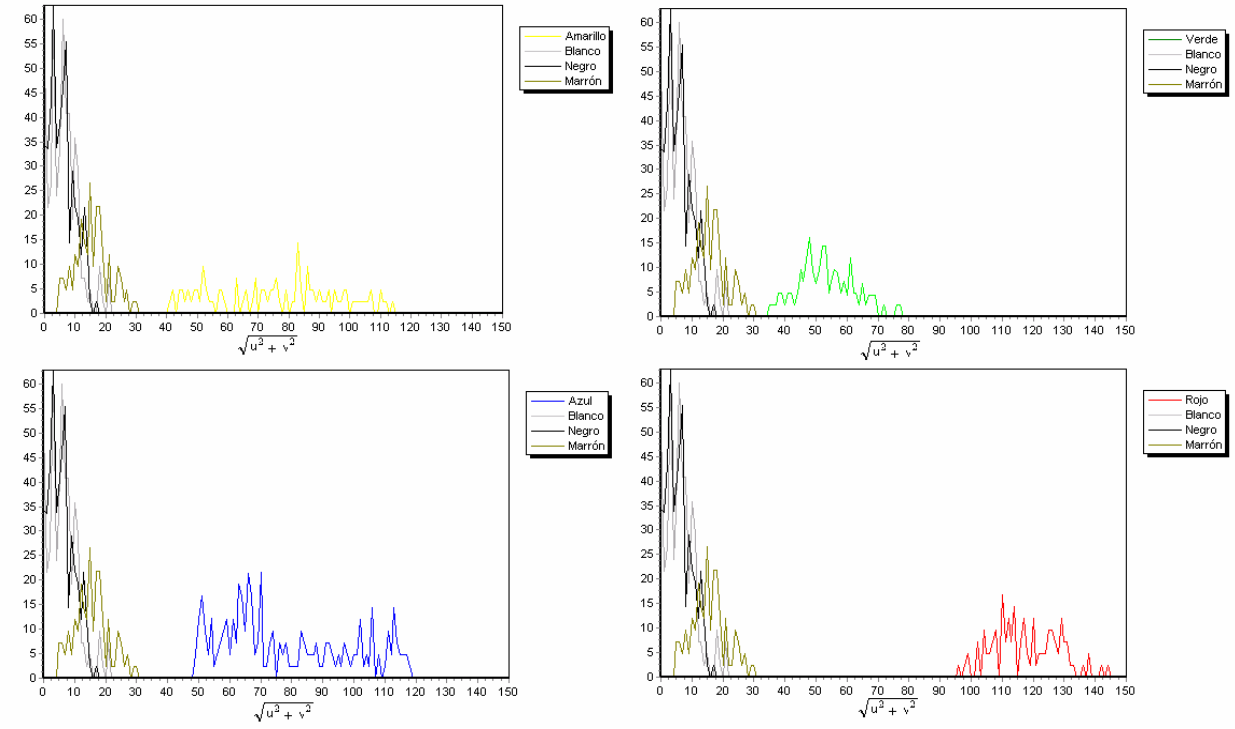


Figura 2. Distribuciones de los valores de $\sqrt{u^2+v^2}$ para los colores del fondo (negro, blanco y marrón) y los diferentes colores de pinturas: amarillo (a), verde (b), azul (c) y rojo (d)

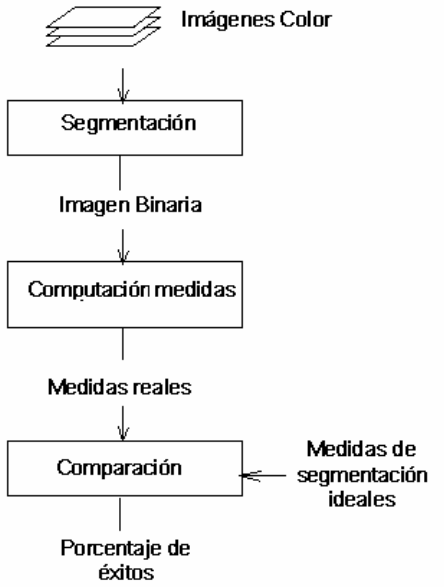


Figura 3. Etapas involucradas en la evaluación de las funciones

3.1 Extracción de características

Las regiones detectadas por el proceso de segmentación son evaluadas según sus características y descriptores de forma. Adquirir mediciones precisas de los objetos es un problema fundamental en análisis de imágenes, ya que tales medidas generalmente brindan información útil a tener en cuenta en la posterior etapa de clasificación [2]. En este caso, los objetos segmentados fueron caracterizados a través de las siguientes métricas:

Área. El área es el momento de orden cero y se calcula como el número de pixels dentro del objeto incluyendo su borde, esto es:

$$A = \sum_x \sum_y I(x, y) \tag{2}$$

Dispersión. Esta métrica indica cuán diseminados se encuentran los pixels del objeto con respecto al centroide. Entonces, para aquellos pixels con $I(x,y)$ =1:

$$D = \sqrt{\sum_x \sum_y \left(x - \bar{x}\right)^2 + \left(y - \bar{y}\right)^2} \tag{3}$$

donde el centroide está dado por los momentos de primer orden $\bar{x}$ y $\bar{y}$:

$$\bar{x} = \frac{\sum_x \sum_y x\ I(x, y)}{\sum_x \sum_y I(x, y)} \qquad \bar{y} = \frac{\sum_x \sum_y y\ I(x, y)}{\sum_x \sum_y I(x, y)} \tag{4}$$

Longitud del eje mayor. El eje mayor de un objeto se define como la línea más larga que puede dibujarse dentro del objeto. Para calcular esta medida, se calculan las distancias entre cada combinación de pixels (x_i, y_i) y (x_j, y_j) sobre el borde del objeto y se determina el par (x_{M1}, y_{M1}) y (x_{M2}, y_{M2}) con la mayor longitud. De esta manera, la longitud del eje mayor es la distancia en pixels entre estos puntos y representa la longitud del objeto. Se calcula como:

$$L_M = \sqrt{\left(x_{M2} - x_{M1}\right)^2 + \left(y_{M2} - y_{M1}\right)^2} \tag{5}$$

Longitud del eje menor. El eje menor se define como la línea más larga que atraviesa al objeto y que puede dibujarse en forma perpendicular al eje mayor. Los puntos extremos del eje menor (x_{m1}, y_{m1}) y (x_{m2}, y_{m2}) se determinan calculando la distancia en píxels entre cada par de puntos del borde (x_i, y_i) y (x_j, y_j) que definen segmentos perpendiculares al eje mayor y determinando el par con la mayor longitud. Así, la longitud del eje menor se obtiene como:

$$L_m = \sqrt{(x_{m2} - x_{m1})^2 + (y_{m2} - y_{m1})^2} \quad (6)$$

3.2 Clasificación por comparación de medidas

En este punto del proceso de evaluación, se ha logrado segmentar uno o más objetos dentro de la imagen de entrada y se han obtenido varias medidas a partir de ellos, de acuerdo a las Ec. (2) a (6). Estas métricas son utilizadas durante la tercera etapa correspondiente a la clasificación como entrada para una regla de decisión que permite determinar si un animal se encuentra en celo.

Con en fin de asistir al proceso de decisión, se pueden usar distintos algoritmos de inteligencia artificial. Sin embargo, en este enfoque se ha implementado una regla sencilla basada en lógica difusa que analiza las diferencias entre las medidas calculadas a partir del conjunto de imágenes de prueba, clasificadas de acuerdo a la evaluación de expertos veterinarios y aquellas obtenidas por el algoritmo para imagen particular (Figura 4). Las medidas obtenidas a partir del primer conjunto fueron usadas como valores de referencia para definir intervalos de confianza para las propiedades geométricas derivadas de la región segmentada (A, D, L_M y L_m). Para determinar un celo positivo se requiere que todas las métricas calculadas a partir de la imagen correspondiente caigan dentro de los respectivos intervalos de confianza. Basado en pruebas experimentales, se ha determinado que pérdidas por debajo del 30% representan casos de no-celo o dudosos, mientras que pérdidas superiores a este porcentaje dan indicio de que el animal está en celo. Por lo tanto, el valor usado para discriminar entre casos de celo positivo y negativo fue definido como $0.3x$, donde x representa cada uno de los descriptores anteriores. El criterio puede ser modificado en el futuro, luego de una retro-información de aplicaciones reales.

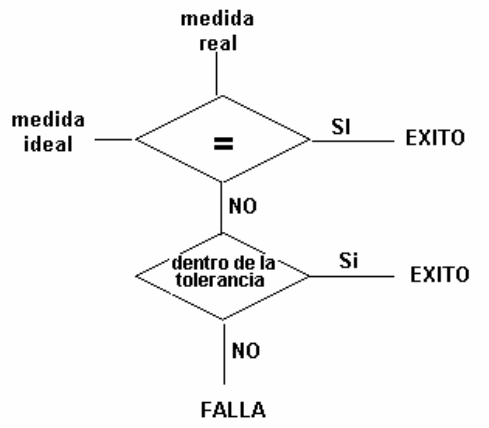


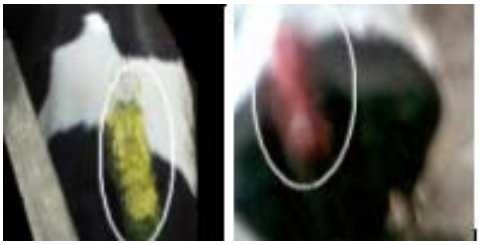
Figura 4. Criterio de éxito

4. VALIDACIÓN EXPERIMENTAL DE FUNCIONES DE UMBRAL

A partir de dos conjuntos de fotografías correspondientes a pinturas sobre cueros vacunos, se llevó a cabo una evaluación integral del método. La primera muestra fue tomada en un ambiente con luz controlada, usando cueros naturales con bandas de pintura de aproximadamente 100 cm^2.

El segundo conjunto de imágenes fue obtenido a partir de fotografías de animales reales en un tambo (El Choique, zona de Napaleofú), a fin de reproducir lo más posible las condiciones de iluminación que se pueden encontrar en aplicaciones reales (Figura 5). La Tabla 1 muestra los resultados obtenidos, indicando el porcentaje de casos que el algoritmo descripto determinó positivos y que fueron confirmados mediante diagnóstico clínico. La segunda columna muestra el porcentaje de éxito de las funciones de umbral seleccionadas para la primera muestra y la tercera columna corresponde al conjunto obtenido a partir de animales reales.

(a)



(b)

Figura 5. Pintadas de color sobre cueros: en muestra artificial (a), ejemplo de muestras con casos reales (b)

Tabla 1. Resultados de las funciones de umbral de color

función de color	% de éxito con fotografías en ambiente controlado	% de éxito con fotografías in situ
$\sqrt{u^2+v^2}$	96.28	94.59
$11\max(\lvert u\rvert,\lvert v\rvert)$	96.85	97.29
$\sqrt{i^2+q^2}$	97.71	97.29
$10\ \max(\lvert i\rvert,\lvert q\rvert)+2\ \min(\lvert i\rvert,\lvert q\rvert)$	97.14	94.59

La Figura 6(a) muestra la performance de las funciones $11\ max(\lvert u\rvert,\lvert v\rvert)$ y $\sqrt{i^2+q^2}$, que lograron los mayores porcentajes de éxito en la clasificación al ser aplicadas a las imágenes del primer conjunto. Como se puede ver, hay un amplio rango de valores de umbral para cada función que permiten obtener buenos resultados de clasificación. Para valores de umbral más pequeños, el algoritmo no alcanza a identificar partes de las pintadas, mientras que en el caso de valores mayores el fondo no se distingue plenamente de las bandas de pintura. La altura de la curva, que corresponde al mayor porcentaje de éxito, indica el poder de segmentación de la función de umbral utilizada, y su ancho determina su robustez (es decir, cuanto más amplio es el rango, menos sensible serán los resultados a las fluctuaciones de color). La Figura 6(b) presenta el comportamiento de las mismas funciones para la segunda muestra de imágenes. En este caso, los resultados parecen menos robustos que en el caso anterior. Sin embargo, hay que notar que esto se debe a que el conjunto obtenido a partir de animales reales es menor que en la muestra controlada.

Aunque la observación visual es el método comúnmente usado para determinar el estado de celo, su eficiencia se estima en sólo 50% en la mayoría de los tambos y se estima que entre el 5 y el 30% de las inseminaciones se realizan en vacas que no se encuentran en celo. El uso de asistentes en el proceso de detección puede mejorar esta situación. Por ejemplo, los sistemas comerciales basados en dispositivos sensibles a la presión reportan una precisión de aproximadamente el 95%. El sistema automático basado en el análisis de pinturas presentado en este trabajo puede brindar una alternativa eficiente para la detección de celo, con una precisión cercana al 97%, siendo mucho más económico y accesible a la mayoría de los productores.

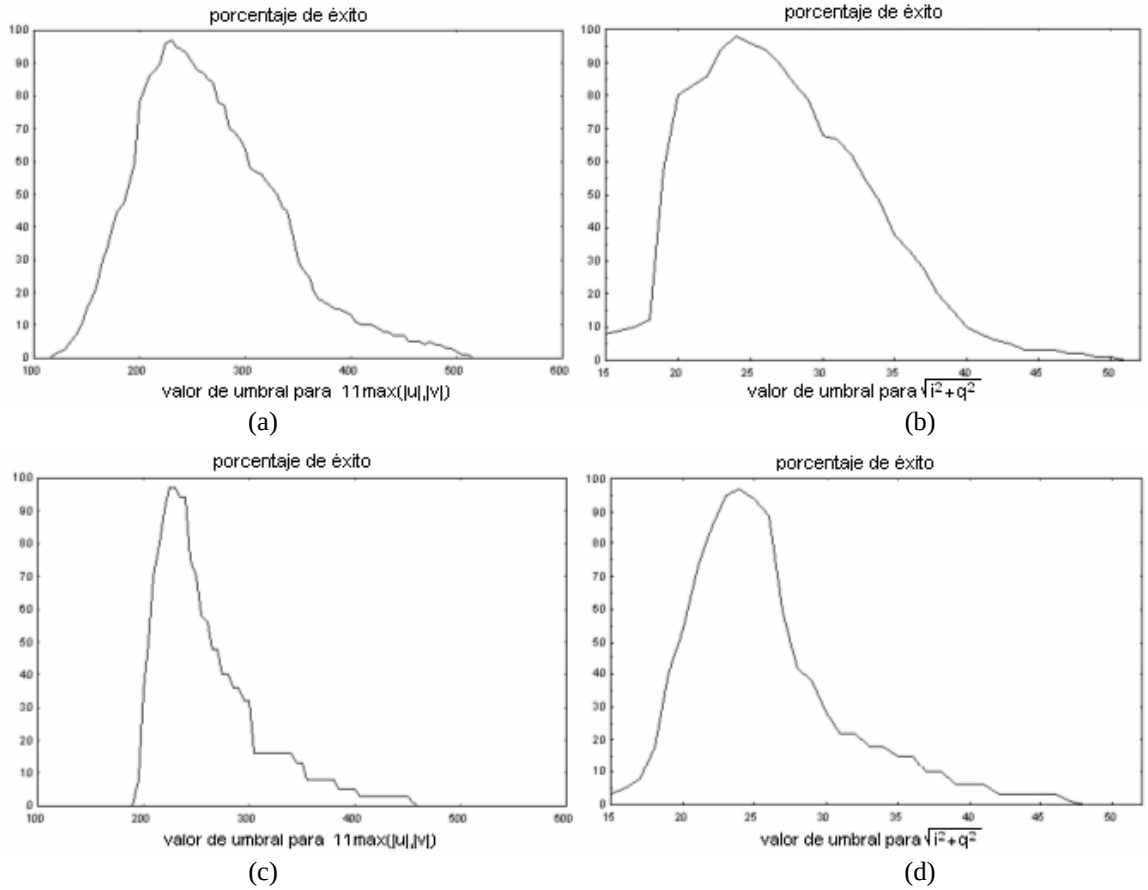


Figura 6. Porcentaje de éxito alcanzado por las funciones $11*\max(|u|,|v|)$ y $\sqrt{i^2+q^2}$ para el primer conjunto de imágenes – laboratorio controlado- (a) y para el conjunto de imágenes tomadas in-situ – tambo-

5. CONCLUSIÓN

Se presentó un método de umbralado para la segmentación automática de imágenes color, basado en transformaciones no-lineales de las componentes de los espacios de color *YUV* y *YIQ*. El método es simple, robusto y ha mostrado una excelente performance con baja complejidad computacional. Además, los valores de umbral y los valores de referencia para la clasificación se deben calcular sólo al comienzo de la operación del sistema. La técnica de segmentación no requiere parámetros de entrada para cada imagen, prescindiendo de la interacción con el usuario.

El algoritmo fue utilizado en una herramienta de soporte de decisiones para detección de celo. Esta herramienta constituye una contribución de alto valor tecnológico para mejorar la eficiencia de este procedimiento, ya que aún no se han desarrollado dispositivos automáticos en este dominio de

aplicación. Los resultados de la segmentación fueron caracterizados mediante medidas geométricas, las cuales brindan un conjunto de parámetros válidos para determinar el estado de celo. La técnica fue probada exitosamente contra conjuntos de fotografías tomadas en ambientes de luz controlada y también en tambos ubicados en el campo. Se ha advertido que ciertos efectos como sombras, cambios de iluminación y ruido pueden afectar la efectividad del proceso de segmentación, y serán investigados en el futuro.

REFERENCIAS

[1] Adams R. and Bischof L., Seeded region growing, *IEEE Trans. Pattern Anal. Machine Intelligence*, 16-6, 641-647, 1994.

[2] Baxes G., *Digital Image Processing: Principles and Applications*, John Wiley & Sons, 1994.

[3] Cheng H., Jiang X., Sun Y. and Wang J., Color Image Segmentation: Advances and Prospects. *Pattern Recogn.*, 34, 2259-2281, 2001.

[4] Gonzalez R. and Woods R., *Digital Image Processing*, Addison-Wesley, 1992.

[5] Haralick R. and Shapiro L., *Computer and Robot Vision*, Addison-Wesley Publishing Company, 1992.

[6] Lucchese L. and Mitra S., Color Image Segmentation: A State-of-the-Art Survey, *Image Processing, Vision, and Pattern Recognition*, The Indian National Science Academy (INSA-A), New Delhi, India, 67-2, 207-221, 2001.

[7] McInerney T. and Terzopoulos D., T-snakes: Topology adaptivee snakes, *Medical Image Analysis*, 4, 73-91, 2000.

[8] Perry G., *Detection of standing estrus in cattle*, http://agbiopubs.sdstate.edu/articles/FS921B.pdf, South Dakota State Univ., 2004.

[9] Pham D., Xu C. and Prince J., A Survey of Current Methods in Medical Image Segmentation, *Annual Review Biomedical Engineering*, 2, 315-337, 2000.

[10] Sezgin M. and Sankur B., Survey over image thresholding techniques and quantitative performance evaluation, *Journal of Electronic Imaging*, 13-1, 146-165, 2004.

[11] Skarbek W. and Koschan A., *Color Image Segmentation: A Survey*, Technischer Bericht, Tech Univ. of Berlin, 94-32, 1994.

Automatic Spot Addressing in cDNA Microarray Images

Mónica G. Larese [†] [*]
Intelligent Systems Group, Instituto de Física Rosario
Rosario, Santa Fe, Argentina
mlarese@ifir.edu.ar

and

Juan C. Gómez [†]
Intelligent Systems Group, Instituto de Física Rosario and
Laboratory for System Dynamics and Signal Processing, FCEIA, UNR
Rosario, Santa Fe, Argentina
jcgomez@fceia.unr.edu.ar

Abstract

Complementary DNA (cDNA) microarrays are a powerful high throughput technology developed in the last decade allowing researchers to analyze the behaviour and interaction of thousands of genes simultaneously. The large amount of information provided by microarray images requires automatic techniques to develop accurate and efficient processing. Each spot in the microarray contains the hybridization level of a single gene. One of the most important features of these images are the regularity and pseudo-periodicity implicit in the spot arrangement. In this paper, an automatic approach based on texture analysis techniques is proposed to localize spots in microarray images. The method estimates the displacement vectors which characterize the texture (*i.e.* the spot arrangement). This is achieved by means of applying the generalized Hough transform on the 2D autocorrelation function previously segmented via morphological operations. The obtained displacement vectors are used to generate a grid template which is matched to the original image. The root mean square error between the estimated locations and the ones computed via a semiautomatic tool is computed to evaluate the accuracy of the process. The method yields promising results with low errors.

Keywords: Bioinformatics, cDNA microarrays, image analysis, automatic addressing.

[*] Author to whom all correspondence should be addressed.
[†] The group is supported by CONICET, ANPCyT under grant PICT 11-15132 and Universidad Nacional de Rosario.

1 INTRODUCTION

Spotted DNA microarrays have become a very useful technology which arised at the middle of the last decade, when the first cDNA microarray was developed by Schena *et al.* [24]. Microarrays allow researchers to compare and analyze thousands of genes simultaneously, and to study their interactions and relations. It also helps to investigate the process of gene expression, by which the information encoded in messenger RNA (mRNA) causes proteins to get synthetized. The fields of usage of DNA microarrays is very wide. The identification of groups of genes involved in the development of certain diseases, in the synthesis of certain proteins and drug design are examples from an extensive list of possible applications.

Spotted cDNA microarrays provide information from two different samples of cDNA which are intended to be compared. A brief explanation on how spotted microarray experiments are carried out is provided next. The reader is referred to [6, 3, 19, 25] for a detailed description.

One of the samples under observation (called the *target sample*) might be taken, for example, from a population composed by individuals suffering from a certain disease (cancer, diabetes, etc.), whereas the other (called the *reference sample*) could belong to a healthy population. The target sample is labeled using red Cy5 dye, and the reference sample using green Cy3 dye. Both samples are mixed and spotted onto a glass containing a control sample bound to its surface. A robot arm with tips at the end is used to print regular arrays (subgrids) of spots on the glass. Usually, in the same glass many subgrids are present, also following an array layout.

After hybridization, the glass is washed and then scanned using different sensors for each one of the two fluorescences, resulting in two digital images (red and green channels). The relative mRNA abundance of both samples can be measured by means of calculating for each spot the Cy5/Cy3 intensity ratio. Due to the low quality of the images, this task is not a trivial one. The reasons for this are the presence of noise, low contrast, non-homogeneous background, artifacts, defective or missing spots, image rotation, spot and subgrid misalignments, among other common problems appearing in such physical processes.

Since thousands of spots per image are generated by each microarray experiment, and usually many images have to be analyzed, the development of automatic algorithms is a crucial issue in order to be able to process large image databases and the huge amount of available information. Automatic processing allows more accurate results and faster computing, and it also prevents from long latency times caused by user interaction.

The tasks involved in the analysis of cDNA microarray images are typically: image preprocessing, gridding (addressing) of the spots, segmentation (extraction of foreground and background intensities for each spot), and measuring the relative mRNA abundance between the samples. The gridding task can also be divided into two stages: the subgrid extraction and the addressing of all the spots in each subgrid. Many works have been published proposing different techniques to develop the gridding task (the reader is referred to [5] for a good review on existing approaches). However, many of them are not completely automatic, requiring user manual intervention to position reference marks on the images. One example is the method proposed by Yang *et al.* [25], which requires the user to select the top-leftmost spot in each subgrid and the bottom-rightmost spot in the bottom-right subgrid to construct a gridding template and to apply it to a batch of images. Another example is the method implemented in the software M.A.G.I.C. [15], where the user needs to select with the mouse the top-leftmost and the top-rightmost spots in a grid and any spot located in the bottom row of it. The selection can be done individually for every grid or the same configuration can be applied for all the grids in the image. The user is then required to manually correct the achieved adressing.

Concerning automatic techniques, many works which are based on the calculation of vertical

and horizontal image intensity profiles, in terms of local or global processing, have been proposed. Regarding local profile processing, the works by Li *et al.* [20] and Blekas *et al.* [7] are worth mentioning. In the case of global profiles, the method proposed by Angulo and Serra [2] implements morphological filtering to process the intensity profiles. In all the previous methods the separation of the subgrids and spots is given by the minimal values of both profiles. One drawback of these techniques is that they fail when the image is rotated, or when the subgrids or spots are misaligned. Many authors have proposed to correct the rotation of the images. However, rotation task has to be done without interpolation to preserve original spot intensities, as proposed by Hirata in [16, 17], where the user manually has to point with the mouse two spots from the same column to calculate the rotation angle. Manually corrected rotation is also proposed by Demirkaya *et al.* in [10]. After doing so, they calculate the vertical and horizontal profiles and autocorrelate them in order to take advantage of spot periodicity and determine the spot spacings.

Several alternatives to compute the angle of rotation automatically have been considered in the literature. For example, Carstensen [8] proposes a gridding method based on a deformable template and Bayesian grid matching, but previously corrects the rotation of the image by means of searching in a restricted range of valid angles in Hough space. Spot spacing is calculated as the distance between two nearest neighbour spots for the corrected angle.

Other methods that also make use of deformable templates, Markov Random Fields and Bayesian matching are those proposed by Katzer *et al.* in [19], and Hartelius and Carstensen in [14]. Ceccarelli and Antoniol [9] implement first a grid matching stage where they estimate the angles between the two grid directions and the x axis from the Orientation Matching (OM) transform. Once they perform that, they project the image in the two directions and calculate the spot spacings from the profiles. After matching an ideal template with the image, they refine it using Bayesian processing.

A data-driven approach has been proposed by Bajcsy in [4], which is based on the optimization of multiple parameters, such as rotation angle, downsampling ratio, number of spots in each row/column per grid, among others. The advantage of the method is that it is fully automatic, but it requires not only much computing time but also to restrict the search space of the parameters to be optimized. On the other hand, Jin *et al.* [18] have implemented a technique based on extended δ-regular sequences.

The authors of the present paper agree with the approaches which consider that it is appropriate to take advantage of the pseudo-periodic patterns and regularity which microarray images exhibit. Even more, the hypothesis stated in this work is that microarray images can be perceived as texture images, and consequently a new approach based on techniques from texture analysis is performed. It is based on image autocorrelation and peak detection. This approach is a variation of the work by Liu *et al.* [23] where it was applied to model periodic patterns by means of frieze and wallpaper groups on general purpose texture images. It also makes use of the algorithm developed by Lin *et al.* [22] to characterize the underlying structure of the pattern by means of the generalized Hough Transform. In the present paper, it is showed how the original method can be adapted to cDNA microarray images in order to estimate two spatial vectors which span an ideal grid template, completely characterizing the pseudo-regular pattern of spots in only one step (angle of rotation from each axis and spot horizontal and vertical spacings). Then, this template is matched to the observed spot centers. Finally, the template is deformed as required to achieve the spot addressing.

The rest of the paper is organized as follows. In Section 2 the proposed techniques are explained in detail. Experimental results obtained after applying the proposed method on a public microarray image database are discussed in Section 3. Finally, concluding remarks and future work are analyzed in Section 4.

2 METHODOLOGY

In this section the proposed techniques for spot addressing are detailed. The whole process is depicted in the block diagram in Figure 1. The raw microarray images used to describe and evaluate the proposed method were taken from the database by Alizadeh *et al.* [1] and correspond to 8-bit depth GIF images.

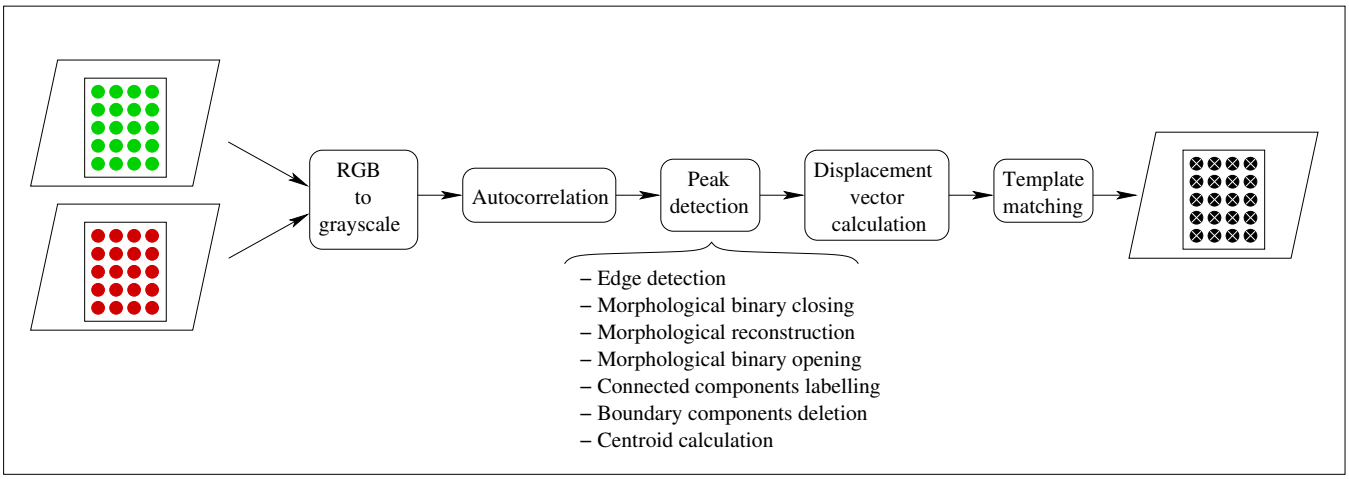


Figure 1: Block diagram for the spot addressing algorithm.

2.1 Spotted microarray image preprocessing

The original microarray images considered in this paper consist of indexed images with an associated RGB (red-green-blue) color map. In order to preprocess the images and address the spots it is necessary to first convert them to grayscale images. It is only an auxiliary step since after the spots are located, the original red and green channels can be used to extract the true intensities.

In order to get the grayscale images, the RGB color model is converted to a YIQ (luminance-hue-saturation) model. This model has the advantage that decouples luminance and chromaticity, codifying in different channels grayscale and color data. To obtain the grayscale information, the I and Q components are set to zero. The Y component is obtained by means of the weighted sum of the R, G and B channels, as described in equation (1) [13]:

$$Y = 0.299R + 0.587G + 0.114B \tag{1}$$

As the B channel is null in cDNA microarray images, the third term of the sum in equation (1) is equal to zero. The result obtained by means of applying the YIQ color model and discarding the I and Q channels has a better contrast than the one which could be obtained by only retaining the I component from the HSI (hue-saturation-value) color model. In the HSI model the I component is the mean value between the R, G and B channels, assigning the same weight to the three components. In the YIQ model, the green component is assigned a higher weight. As for equal amounts of red and green intensities red is more strongly perceived by humans, the YIQ model achieves a result with improved contrast after the I and Q components are discarded.

2.2 Image autocorrelation

Autocorrelation was proposed as a means of characterizing regular texture structures (extraction of texture primitives and displacement vectors which describe the spatial arrangement of the primitives) by Lin *et al.* [22], since, as it is well known, the peaks on the autocorrelation function show the same periodicity as the original image. Autocorrelation is also considered more robust than Fourier transform since the peaks are stronger and consequently simpler to find. Moreover, Lin *et al.* state that autocorrelation requires less computing time than co-ocurrence matrices [26], achieving more accurate results.

In the present work the normalized centered image autocorrelation is computed on the grayscale image. Autocorrelation is defined as the correlation of an image $f(x, y)$ with itself, and it is mathematically described in equation (2):

$$f(x,y) \circ f(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m,n) f(x+m, y+n) \tag{2}$$

where $f^*(x, y)$ is the complex conjugate of $f(x, y)$ and the symbol '$\circ$' stands for the correlation operator.

2.3 Peak detection

Since the peaks on the autocorrelation function are periodically spaced, it is possible to restrict the analysis to a small portion of the function containing only a few periods in each one of the two axis directions. In this work, the subregion was 301×301 pixel sized and it was taken from the center of the autocorrelation function. However, the size of this window should not be a highly sensitive parameter, since the only restriction is that it must contain a few periods of the autocorrelation function.

Different approaches have been proposed to detect maximal points in the autocorrelation. The one proposed in [22] consists of applying a filter that automatically smoothes the autocorrelation surface to eliminate irregularities and easily detect the local maxima. In the work by Liu *et al.* [23] preliminary candidate peaks are obtained by means of non-maximal suppression. However, the positions of the peaks correspond to discrete coordinates, since it is a discrete 2D autocorrelation function. This issue yields discrete coordinates for the displacement vectors to be found in the subsequent steps (see Section 2.4), propagating the errors to the estimated lattice. In the present paper, a new approach is presented to extract the candidate peaks using continuous coordinates. It improves the calculation of the two displacement vectors, making it possible to find vectors with continuous components. The idea of the approach if to find the centres of the segmented components in the autocorrelation image, and it consists of the following steps:

1. **Edge detection**: the edges in the 2D autocorrelation function are extracted by first applying a 13×13 LoG edge detector with standard deviation $\sigma = 2$ and then looking for the zero-crossings [12].

2. **Morphological binary closing**: on the resulting edge image a morphological binary closing is implemented using a circular structuring element of radius equal to 3 pixels. The reason for this is to fill any possible gaps existing on the recognized edges.

3. **Morphological reconstruction**: holes inside object boundaries are filled by means of morphological binary reconstruction. Let $f(x, y)$ be the binary image after edge detection on the autocorrelation image, whose holes are wished to be filled. Let the complement of $f(x, y)$ be

$f^C(x, y)$. Suppose $g(x, y)$ is another binary image, whose pixels are all equal to zero except for those which are located on the image border, where their value is $1 - f(x, y)$. It is verified that $g(x, y) \subseteq f(x, y)$, where $g(x, y)$ is the marker image (initial stage for the reconstruction) and $f^C(x, y)$ is the mask (constrain for the reconstruction). Computing the morphological binary reconstruction with the mask and marker images stated above yields the filling of the holes [13] delimited by the edges detected previously.

4. **Morphological binary opening**: a morphological binary opening is developed using a circular structuring element with radius equal to 5 pixels. This is performed to break appart components which are touching one to each other.

5. **Connected components labelling**: the connected componentes are labelled in the segmented autocorrelation image.

6. **Boundary components deletion**: the components touching the border of the autocorrelation image are deleted because if they are not completely contained in the image they would introduce errors in the centroids calculation.

7. **Centroid calculation**: the centroids of the remaining components are then computed and the coordinates of these centroids are considered as the candidate peaks of the autocorrelation function.

Once the procedure described above is concluded, the approach based on regions of dominance proposed in [23] is followed to determine the prominent candidate peaks, as it is explained in the next paragraphs.

The underlying idea is that the height of a peak is not perceived as important as how far it is from a higher peak, that is to say, how extensive the peak's circular region of dominance is. In order to determine which are the most important peaks according to this concept, the candidate peaks are first sorted in descendent order of height. The next step consists of calculating the distance from each peak to each one of its predecesors in the ordered list. For each candidate peak, the distance to its nearest higher peak (extension of its region of dominance) is recorded. Then all the peaks are reordered from the largest to the smallest region of dominance. The next step consists of using the candidate peaks to find the displacement vectors which characterize the lattice on which the spots are located. This procedure is explained in the following section.

2.4 Finding the displacement vectors

In order to find the two vectors that generate the lattice, special care must be taken. The reason is that the vectors not necessarily are the shortest ones, due to the possible presence of spurious peaks. Neither they are the largest ones, because probably they could be larger scaled versions of the desired vectors. Then, the goal is to find the shortest displacement vectors satisfying linear independency which are able to generate the whole lattice.

Lin *et al.* [22] proposed a procedure based on the generalized Hough transform to successfully find the lattice vectors, and was also successfully applied in [23] with small variations. The use of the generalized Hough transform is advantageous in the sense that it is robust to noise in the regular pattern of the autocorrelation surface. In the present paper the original procedure was implemented, as detailed below.

The ordered peaks found in Section 2.3 are regarded as vectors whose components are their cartesian coordinates. All of them are then translated to set the axis origin on the first peak (vector) of the

list. As a consequence, the vector with the largest region of dominance becomes $v_0 = (0, 0)$. The procedure to find the displacement vectors that generate the lattice starts by initializing a 2D accumulator array to zero. This array is square and has as many elements per side as vectors are to be considered. For each pair of linearly independent vectors (taken in order from the arranged list), their linear combinations are computed. At the same time, for each one of the remaining vectors, if they lye near any vertex of the lattice spanned by the pair of linear independent vectors, the accumulator is increased with a high score in the cell corresponding to the pair of spanning vectors which are being considered. Otherwise, the incremental score is lower. The procedure is iteratively repeated for all the vector pairs. Finally, the pair of vectors which have the highest score in the 2D accumulator array are retained.

After the two vectors have been selected, they are used to calculate the sum and the difference vectors. Among the four of them (the two original vectors, the sum and the difference vectors) the two ones with the lowest 2-norm are regarded as the displacement vectors which generate the spot lattice. They together characterize the vertical and horizontal spot spacings, the angle of rotation of the image and the directionality of the columns and rows of spots.

2.5 Template matching

Once the two displacement vectors that generate the ideal spot lattice have been estimated, a template grid can be constructed. The numbers M and N of rows and columns, respectively, are known a priori from the microarrayer configuration. It is desirable that the starting point for the template coincides with the top-leftmost spot of the real subgrid. In order to detect the location of this spot, the original image is temporarily corrected for rotation and then the two vertical and horizontal smoothed profiles are computed on the corrected image. The most prominent M and N peaks are found on these profiles, but only the coordinates of the one which spatially appears first in each profile are retained. These coordinates are then back-rotated to the original position of the image, yielding the location of the top-leftmost spot in the original subgrid from which the template can be spanned.

On the other side, the spot centres on the original microarray image are calculated. In order to achieve this, a variation of the procedure used to segment the autocorrelation function (previously described in Section 2.3) is used. First, the edge borders are computed on the original microarray image by means of the LoG edge detector followed by zero-crossing localization. Then the holes are filled through morphological binary reconstruction. Next, morphological erosion is applied using a disk with radius equal to 3 pixels as a structuring element. Finally, the component centroids are detected on the resulting binary image, yielding as a result points which ideally should correspond to the observed spot centres. However, due to the typical characteristics of these kind of physical experiments, artifacts that are present in the image cause the detection of spurious centres, and in many cases overlapped spots cause only one centroid to be detected in the middle of two spots. Moreover, the centres corresponding to missing spots or spots with low contrast can not be detected at all. For this reason, the computed centres are matched to the template generated in the previous step. This processing stage discriminates spurious from proper centres, providing also a way to adjust the position of the spots of the template with little distortions on it. To achieve this, for each spot in the template a local search is developed inside a square box around it to detect if there exists any observed spot centre within its limits. If this is true, the position of the template spot is replaced by the coordinates of the observed spot centre. Otherwise, the template spot is left unchanged.

Experimental results obtained after implementing the described procedure are detailed in the following section.

3 EXPERIMENTAL RESULTS

In this section the performance of the proposed algorithm is evaluated on real microarray images from a public database from the Lymphoma/Leukemia Molecular Profiling Project described in [1]. The images are available for free download from http://llmpp.nih.gov/lymphoma/data/rawdata. In Figure 2a. one of the subgrids of a microarray image is shown. The image has been artificially rotated 5 degrees to show the performance of the algorithm when correction for rotation is required. The goal consists of addressing all the spots in the subgrid, including those which are missing. As can be seen from Figure 2a., the subgrid exhibits a pseudo-periodic regular pattern. This periodicity and regularity is even more visible in the autocorrelation image shown in Figure 2b.

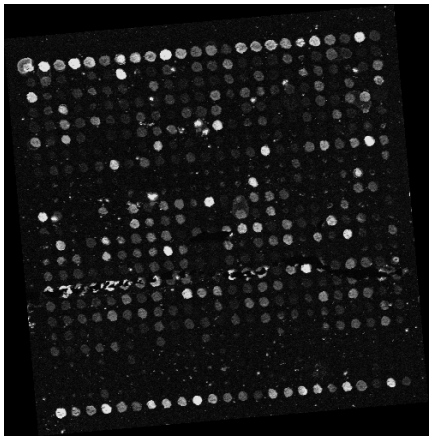
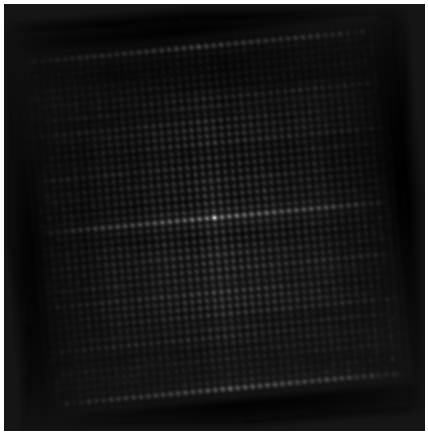
a.  b. 

Figure 2: a. Original subgrid microarray image. b. Image autocorrelation.

Only a few periods of the autocorrelation function are necessary to develop the texture structure analysis. Figure 3a. shows the extracted 301×301 pixel sized portion taken from the central part of the autocorrelation function. A 3D version of the same function is depicted in Figure 3b. On this subimage the segmentation procedure is developed in order to extract the centres of the connected components and regard them as the candidate peaks for texture characterization. The result of applying this procedure to the image in Figure 3a. is included in Figure 3c. In this Figure, all of the components which were touching the image border were removed. The red crosses indicate the position of the centroids which have been calculated for each remaining component.

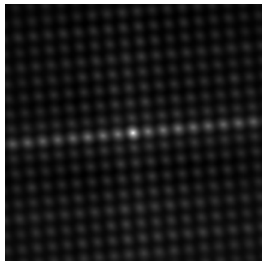
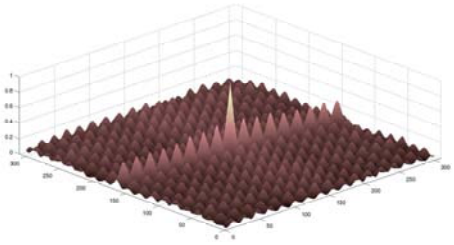
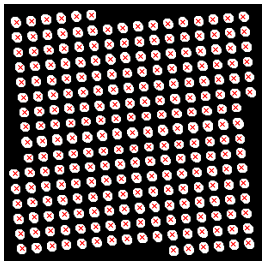
a.  b.  c. 

Figure 3: a. Central portion extracted from the autocorrelation image. b. Central portion extracted from the autocorrelation image as a 3D function. b. Segmented autocorrelation with connected component centres in red.

The displacement vectors that describe the structural information of the texture and span the spot lattice are: $v_1 = (17.62, 1.64)$ and $v_2 = (1.37, -17.60)$. From these vectors the angle of rotation of the image is estimated in 5.31 degrees. It is the angle that the vector with the largest slope forms with the horizontal axis. Aditionally, the angle formed by the two vectors especifies the existing angle between each row and each column of spots in the grid. The horizontal and vertical spot spacing is given by the length of each vector, and they are 17.69 and 17.66 pixels, respectively. The original image and the same image corrected for rotation are shown in Figures 4a. and 4b., respectively.

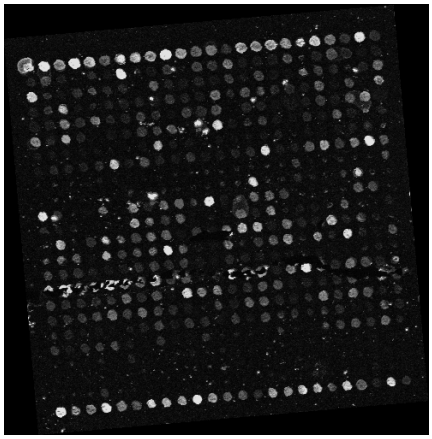
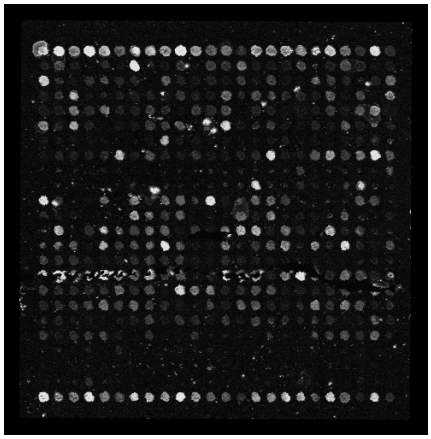
a.  b. 

Figure 4: a. Original subgrid microarray image. b. Original image corrected for rotation.

In order to determine the top-leftmost spot from which to construct the grid template, smoothed versions of the horizontal and vertical profiles are computed from the subgrid corrected for rotation. The resulting profiles calculated on the image in Figure 4b. are depicted in Figure 5a. and 5b., respectively, where the $M = 24$ and $N = 24$ maximum peaks corresponding to the total number of rows and columns of spots are marked with red crosses. Only the coordinates of the first detected peak is recorded from each profile. These coordinates are then back-rotated -5.31 degrees to recover the true position in the original image.

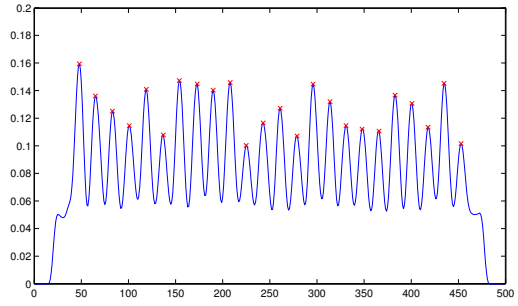
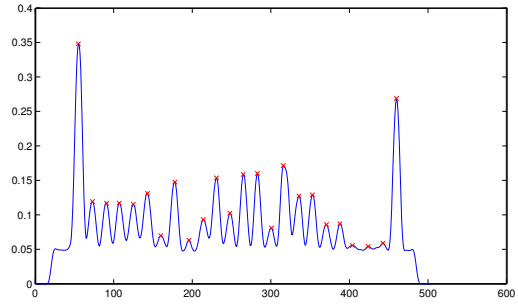
a.  b. 

Figure 5: Smoothed intensity profiles of the image corrected for rotation (local maxima are shown in red). a. Horizontal profile. b. Vertical profile.

After applying the procedure described above, the template grid is generated. Figure 6a. shows separating lines over the original image representing the grid spanned by the two displacement vectors v_1 and v_2. The red crosses indicate the observed spot centres calculated by means of segmenting the

original microarray image using the steps detailed in Section 2.5. Figure 6b. shows the final result, after matching the ideal spot centres of the template to the observed spot centres. Green circles indicate the position corresponding to the estimated template. Note also how spurious spot centres corresponding to noise and image artifacts have been discarded.

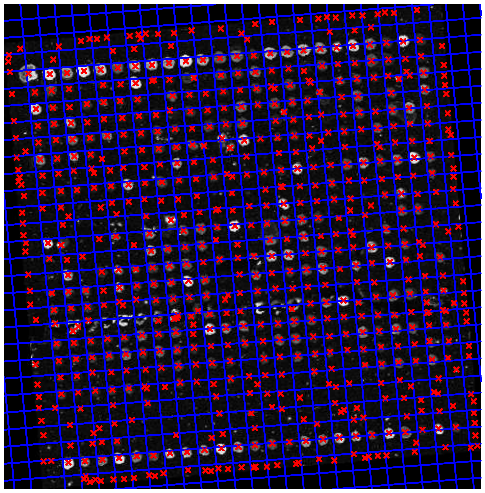
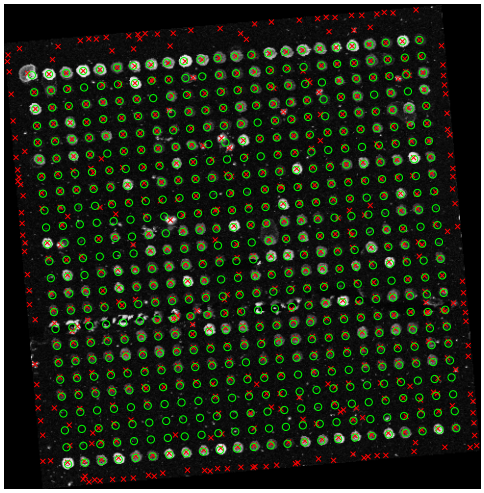
a.  b. 

Figure 6: a. Separating lines delimiting texture primitives (spots). b. Template spots overlapped on the observed subgrid image (red crosses indicate observed spot centres and green circles are template spots.

The whole procedure was applied to different real microarray images from the dataset of Alizadeh *et al.* [1]. This dataset also provides the addressing results obtained by processing them with the semi-automatic software tool ScanAlyze 2.3 [11]. The Root Mean Square Error (RMSE) was computed between the estimated spot centres addressed using the automatic method proposed in this work and the true spot centres calculated with ScanAnalyze. It is noticeable that as ScanAlyze is a semiautomatic tool, there may be mistakes introduced by user selection. The results are reported in Table 1.

Table 1: RMSE (in pixels) between the estimated location of the spot centres using the proposed automatic method and the positions obtained with the semiautomatic tool ScanAlyze.

Image ID	Number of spots	RMSE in x	RMSE in y	Total RMSE
lc8n015rex2	18432	3.43	8.35	9.03
lc7b104rex2	9216	3.49	3.85	5.20
lc7b023rex2	9216	3.42	3.77	5.09
lc7b017rex2	9216	3.77	4.73	6.05
lc7b046rex2	9216	3.34	4.08	5.27
lc4b063rex2	9216	5.75	4.98	7.61

The proposed approach detects the position of the spot centres with a low error. However, it would be necessary to improve the estimation by means of a refinement procedure which allows to diminish the error. At the moment, proofs are being made using Markov Random Fields (MRF) [21].

4 CONCLUSIONS AND FUTURE WORK

In this paper an automatic approach is proposed to address the location of microarray subgrid spot centres. It relies on the assumption that spotted microarray images can be regarded as texture images and consequently texture analysis techniques are suitable to be applied. This is because of the regularity and pseudo-periodicity exhibited by microarray images.

The present approach computes the displacement vectors that span the spot lattice, finding with a single technique the image angle of rotation and the row and column spot spacing. These approach is based on the computation of the generalized Hough transform with the candidate peaks ordered according to their region of dominance in the autocorrelation function. Instead of using the raw autocorrelation image, this image is previously segmented by means of morphological binary operations and connected components detection. The centres of these components are computed to get continuous coordinates for the peaks and consequently, continuous cartesian components for the displacement vectors.

A grid template is generated from the two displacement vectors and then it is matched to the original spot centres. The RMSE was calculated between the estimated spot locations and the ones obtained by the semiautomatic tool ScanAnalyze in order to evaluate the performance of the procedure. The method yields promising results. However, a refinement procedure would be desirable to reduce the estimation error improving accuracy. This procedure is currently under development.

REFERENCES

[1] Alizadeh A. A., Eisen M. B., *et al.* Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, February 2000.

[2] Angulo J. and Serra J. Automatic analysis of DNA microarray images using mathematical morphology. *Bioinformatics*, 19(5):553–562, 2003.

[3] Attwood T. K. and Parry-Smith D. J. *Introduction to bioinformatics*. Addison Wesley Longman Limited, Harlow, England, 1999.

[4] Bajcsy P. Gridline: Automatic grid alignment in DNA microarray scans. *IEEE Transactions on Image Processing*, 13(1):15–25, 2004.

[5] Bajcsy P. An overview of DNA microarray grid alignment and foreground separation approaches. *EURASIP Journal on Applied Signal Processing*, Article ID 80163:1–13, 2006.

[6] Berrar D. P., Dubitzky W., and Granzow M., editors. *A practical approach to microarray data analysis*. Kluwer Academic Publishers, Dordrecht, 2003.

[7] Blekas K., Galatsanos N. P., Likas A., and Lagaris I. E. Mixture model analysis of DNA microarray images. *IEEE Transactions on Medical Imaging*, 24(7):901–909, 2005.

[8] Carstensen J. M. An active lattice model in a Bayesian framework. *Computer Vision and Image Understanding*, 63(2):380–387, 1996.

[9] Ceccarelli M. and Antoniol G. A deformable grid-matching approach for microarray images. *IEEE Transactions on Image Processing*, 15(10):3178–3188, 2006.

[10] Demirkaya O., Asyali M. H., Shoukri M. M., and Abu-Khabar K. S. Segmentation of microarray cDNA spots using MRF-based method. the 25th Annual Conference of the IEEE EMBS, Cancun, Mexico, 2003.

[11] Eisen M. Scanalyze. 1999. http://rana.lbl.gov/EisenSoftware.html.

[12] Gonzalez R. and Woods R. *Digital image processing*. Prentice Hall, 2nd. edition, 2002.

[13] Gonzalez R. C., Woods R. E., and Eddins S. L. *Digital image processing using MATLAB*. Prentice Hall, 2004.

[14] Hartelius K. and Carstensen J. M. Bayesian grid matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):162–173, 2003.

[15] Heyer L. J., Moskowitz D. Z., Abele J. A., Karnik P., Choi D., Malcolm Campbell A., Oldham E. E., and Akin B. K. MAGIC tool: integrated microarray data analysis. *Bioinformatics*, 21(9):2114–2115, 2005.

[16] Hirata Jr. R., Barrera J., Hashimoto R. F., and Dantas D. O. Microarray gridding by mathematical morphology. In *Proc. SIBGRAPI, Florianópolis*, pages 112–119. IEEE, 2001.

[17] Hirata Jr. R., Barrera J., Hashimoto R. F., Dantas D. O., and Esteves G. H. Segmentation of microarray images by mathematical morphology. *Real-Time Imaging*, 8:491–505, 2002.

[18] Jin H.-J., Chun B.-K., and Cho H.-G. Extended δ-regular sequence for automated analysis of microarray images. *EURASIP Journal on Applied Signal Processing*, Article ID 13623:1–11, 2006.

[19] Katzer M., Kummert F., and Sagerer G. Methods for automatic microarray image segmentation. *IEEE Transactions on Nano-Bioscience*, 2(4):202–214, 2003.

[20] Li Q., Fraley C., Eugene Bumgarner R., Yeung K. Y., and Raftery A. E. Donuts, scratches and blanks: robust model-based segmentation of microarray images. *Bioinformatics*, 21(12):2875–2882, 2005.

[21] Li S. Z. *Markov Random Field modeling in computer vision*. Springer-Verlag, 1995.

[22] Lin H.-C., Wang L.-L., and Yang S.-N. Extracting periodicity of a regular texture based on autocorrelation functions. *Pattern Recognition Letters*, 18:433–443, 1997.

[23] Liu Y., Collins R. T., and Tsin Y. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):354–371, 2004.

[24] Schena M., Shalon D., Davis R. W., and Brown P O. Quantitative monitoring of gene expression patterns with a complementary cDNA microarray. *Science*, 270:467–470, 1995.

[25] Yang Y., Buckley M., Dudoit S., and Speed T. Comparison of methods for image analysis on cDNA microarray data. Technical Report #584, Department of Statistics, University of California, Berkeley, November 2000. Document available at URL: http://www.stat.berkeley.edu/users/terry/zarray/TechReport/584.pdf.

[26] S. W. Zucker and D. Terzopoulos. Finding structure in cooccurrence matrices for texture analysis. *Computer Graphics and Image Processing*, 12(3):286–308, 1980.

Desarrollo de un escáner 3D mediante cámaras estereoscópicas e iluminación láser

Federico Cristina, Sebastián H. Dapoto, Javier Vegas, Verónica Artola, Claudia Russo, Abásolo María José
{fcristina, sdapoto, jvegas, vartola, crusso} @lidi.info.unlp.edu.ar
{mjabasolo} @uib.es

Instituto de Investigación en Informática LIDI
Facultad de Informática. Universidad Nacional de La Plata.
La Plata, Buenos Aires, Argentina.
Universidad de las Islas Baleares
Departamento de Matemática e Informática

V Workshop de Computación Gráfica, Imágenes y Visualización
XIII Congreso Argentino de Ciencias de la Computación CACIC 2007

Resumen

Los dispositivos de escaneo tridimensional permiten obtener modelos de objetos utilizando distintas técnicas de captura. Esta tarea puede ser llevada a cabo por ejemplo mediante estereovisión, el cual es un método de reconstrucción 3D a partir de fotografías. Las técnicas de reconstrucción 3D mediante luz se basan en la proyección de un patrón de luz conocido sobre una escena y a partir del análisis de la proyección puede deducirse la forma de los objetos. De esta manera, basándose en la información bidimensional de las fotografías en conjunto con la luz estructurada puede obtenerse la ubicación tridimensional y construirse un modelo virtual 3D de la escena u objeto fotografiado. El presente trabajo tiene por objeto plantear una solución integral hardware/software al problema de reconstrucción de modelos tridimensionales que presente un buen balance costo/beneficio.

Palabras claves: estereovisión, escáner tridimensional, reconstrucción 3D, proyección de luz

Abstract

Three-dimensional scanning devices allow obtaining object models using several capturing techniques. This task can be carried out through stereovision, which is a 3D reconstructing method based on photographies. 3D reconstruction techniques using light projection uses a known light pattern over the scene, the shape of the objects can then be deduced by the analysis of the projection. In this way, based on the two-dimensional information from the photographies together with the structured light, the three-dimensional position can be obtained and build a 3D virtual model of the photographed object or scene. The present paper poses an integral hardware/software solution to the three-dimensional model reconstruction problem which presents a good cost/benefit balance.

Keywords: stereovision, three-dimensional scanner, 3D reconstruction, light projection

[1] Esta investigación es financiada por la Agencia Española de Cooperacion Internacional AECI programa de cooperación interuniversitaria e investigación científica. Proyecto A/7155/06 – Diseño de un sistema de reconstrucción 3D mediante cámaras estereoscópicas y luz estructurada.

1 INTRODUCCIÓN

En la actualidad existen distintos sistemas de digitalización 3D [1,2,3,4,5,6]. Según sus características, éstos pueden ser divididos en dos grandes grupos: sistemas con o sin contacto con el objeto a digitalizar.

1.2 Escaneado por contacto

Los sistemas de digitalización por contacto son los más antiguos. Principalmente se emplean en la verificación dimensional de piezas industriales para el control de calidad. Con estos sistemas se obtienen las coordenadas de los puntos gracias al desplazamiento de una punta sobre la superficie a digitalizar. Éstos poseen una elevada precisión; pero por el contrario tienen una velocidad de adquisición de datos muy baja, ya que se necesita llevar manualmente la punta a cada posición que se quiera digitalizar, con lo cual el tiempo de escaneado de un objeto de tamaño medio resulta relativamente elevado. Para emplear estos sistemas por contacto, se necesita además que las piezas tengan la rigidez suficiente para que no se deformen por el contacto de la punta y debido a la geometría de las éstas, es imposible digitalizar ciertas ranuras y ángulos interiores.

1.2 Escaneado sin contacto

Los sistemas de digitalización sin contacto presentan la ventaja de lograr una velocidad de adquisición de datos muy superior a las de los digitalizadores por contacto. Se pueden dividir las técnicas de digitalización sin contacto en dos grandes grupos: de visión pasiva y de visión activa.

1.2.1 Técnicas de visión pasiva

El sistema visual humano permite obtener información de profundidad mediante la fusión de dos escenas monoculares, que son las escenas que captan cada uno de nuestros ojos. La visión pasiva se basa entonces en utilizar dos puntos de vista de un mismo objeto para encontrar las coordenadas tridimensionales.

La principal complejidad de este método es la correspondencia de puntos en cada una de las imágenes monoculares. Las técnicas encargadas de realizar esta tarea presentan generalmente un elevado costo computacional [7,8].

1.2.2 Técnicas de visión activa

Estás técnicas son las que hacen intervenir una fuente de luz específica para determinar las coordenadas tridimensionales de los puntos de medida. Constan como mínimo de un emisor de luz y uno o más receptores. Conociendo la dirección del rayo emitido y la del recibido se obtienen las dimensiones del triángulo formado y por lo tanto se logra determinar la profundidad del punto inspeccionado.

Existen tres tipos de sistemas de visión activa:

- Telemetría: La telemetría consiste en medir el tiempo de recorrido de un rayo luminoso (láser) hasta la superficie de medida. La medición se obtiene determinando el tiempo entre la emisión del impulso luminoso y la observación del retorno.

- Proyección de luz estructurada: En este sistema el emisor es un proyector de luz blanca y el receptor una cámara CCD. Cuando se inicia una digitalización el proyector lanza sobre el objeto una serie de franjas de luz verticales de claros y sombras alternadas, que son registradas por la cámara. El cálculo de la profundidad consiste en resolver las intersecciones plano-recta de la proyección.

- Digitalización por láser: Este sistema, utilizado en el presente trabajo, utiliza un láser de diodos como fuente lumínica. Dicho láser proyecta una línea de luz sobre la superficie que va a ser digitalizada. La luz reflejada será detectada por uno o dos células fotosensibles que se encuentran situadas a ambos lados del láser. Estos detectores leen el haz de luz reflejado y procesan la información obtenida a partir del perfil proyectado. El resultado de la digitalización nos da la posibilidad de obtener la geometría completa de la pieza a escanear.

Aunque cada uno de estos sistemas posee características similares, éstas presentan ventajas y desventajas de acuerdo a los requerimientos específicos de la aplicación, las cuales serán comentadas a continuación.

2 ANALISIS PRELIMINAR DE SISTEMAS

La medición 3D de geometrías libres ha sido abordada mediante el desarrollo de instrumentos de medición como Máquina de Medir por Coordenadas (MMC), Sistemas Láser de Medición (SLM) y Sistemas de Medición por Visión (SMV).

Los SLM son utilizados en la medición de longitud y ángulo con alta exactitud. Su desventaja radica en el empleo de arreglos ópticos que generalmente son diseñados para evaluar el desempeño de máquinas y herramientas de ejes coordenados.

Por otra parte, existen SLM diseñados para la medición de profundidad en escenas con una alta tasa de muestreo. Su desventaja radica en la relativa baja exactitud. En forma adicional, los instrumentos comerciales ya sea MMC o SLM tienen un alto costo de compra y mantenimiento, lo que hace difícil su disponibilidad en el ámbito nacional.

Por el contrario, los SMV constituyen una oferta atractiva en cuanto a costos, a expensas de la disminución en exactitud. En forma adicional, los SMV pueden tener las más altas tasas de muestreo y los niveles más elevados de automatización en su operación. No obstante las desventajas de los SMV, existen aplicaciones industriales, artísticas y de entretenimiento que los convierten en una opción ideal.

3 DESCRIPCION DE LA TECNICA IMPLEMENTADA

Básicamente, la técnica emplea un sistema de medición sustentado en la generación de un patrón con luz proyectada y la captura de imágenes estéreo. La proyección resalta las características de interés en el objeto a medir y facilita el procesamiento de la imagen para la reconstrucción tridimensional como también simplifica y optimiza el proceso de correspondencia de puntos utilizado en el procesamiento de las imágenes.

3.1 Metodología de adquisición

3.1.1 Tipos de proyecciones

El haz láser puede ser conformado de diferentes formas (figura 1): punto, una o más líneas, cruz, círculo, grillas de diferentes pasos y tamaños [9,10,11].

Figura 1. Varios tipos de proyecciones mediante láser

Las formas más utilizadas son el punto, la línea y la cruz. Las diferentes formas del haz láser se utilizan en alineación, centrado, marcado de línea de corte, inspección, seguridad, visión industrial. Se determinó la línea como el patrón correcto para el sistema.

3.1.2 Técnica implementada

El sistema fue pensado a fin de incorporar dos métodos de adquisición de datos diferentes, los cuales comparten características similares.

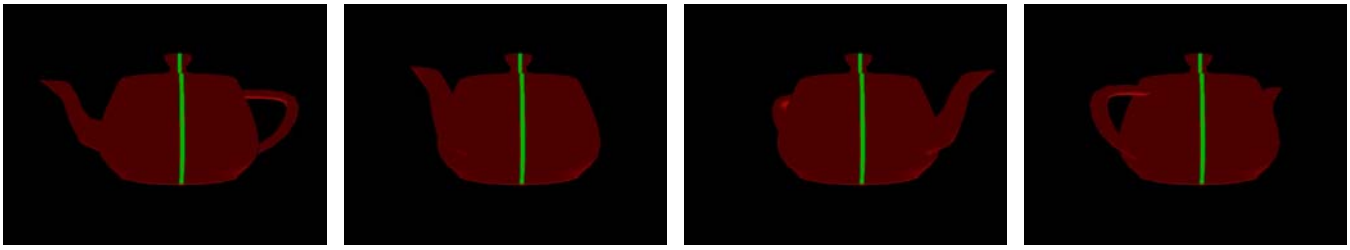
- Adquisición de datos mediante barrido láser: en este caso, el objeto a reconstruir permanece inmóvil, mientras que se releva la superficie del mismo barriendo el área de escaneado con un láser. Pensado para objetos sin volumen de revolución o cuyo contenido se presenta en una sola cara.

- Adquisición de datos mediante objeto giratorio: en este caso, el láser permanece inmóvil, y es el objeto quien realizar un giro sobre su eje a fin de relevar la superficie del mismo. Ideal para objetos con volumen de tipo *cilíndrico*, como vasijas, tazas, etc.

Las simulaciones ilustradas en las figuras 2 y 3 muestran una secuencia de imágenes que ejemplifican la captura de ambos mecanismos. La primera presenta la adquisición mediante barrido láser, mientras que la segunda presenta la secuencia mediante objetivo giratorio.

Frame 5 Frame 15 Frame 25 Frame 35

Figura 2. Barrido láser (total 50 frames)



Frame 0 Frame 27 Frame 71 Frame 108

Figura 3. Objetivo giratorio (total 180 frames)

3.1.3 Detalle del proceso de reconstrucción

Primeramente, y por única vez, es necesario realizar la calibración de cámaras, tanto intrínseca como extrínsecamente [12,13,14,15,16,17]. El sistema presenta un módulo de calibración específico para esta finalidad.

Posteriormente, una vez obtenidas las capturas estéreo, se procede a realizar la reconstrucción del objeto mediante la obtención de la nube de puntos. Básicamente se genera una matriz de profundidades, en la que cada celda (X, Y) refiere a un Z del objeto en particular. La columna *i-esima* de la matriz se obtiene con el par de imágenes estéreo *i-ésimo* de la secuencia de video, mediante la triangulación de los puntos iluminados con la proyección del láser.

Por cada par de imágenes estéreo, se debe realizar la correspondencia de puntos en la zona donde se presenta la proyección del láser. Gracias a ésta, dicha tarea es relativamente sencilla y presenta un reducido coste computacional.

El pseudocódigo para la obtención de la matriz de profundidades es el siguiente:

```
float depths[][];
int column = -1;
for (int frame = 0; frame < leftVideo.frames; frame++)
{
       leftImage  = leftVideo.getFrame(frame);
       rightImage = rightVideo.getFrame(frame);

       column++;
       for (int leftRow = 0; leftRow < leftImage.rows; leftRow++)
       {

              rightRow = obtainRowCorrespondence(leftRow);

              leftColumn  = obtainLaserPosition(leftImage,  leftRow);
              rightColumn = obtainLaserPosition(rightImage, rightRow);

              zValue = triangulate(leftColumn, leftRow, rightColumn, rightRow);
              depths[column][leftRow] = zValue;
       }
}
```

La matriz obtenida es posteriormente utilizada para la renderización en el entorno de visualización 3D. En el caso del objeto giratorio, la reconstrucción del objeto se realiza de manera tal que se contemple el ángulo de giro en cada columna a reconstruir.

4 PRUEBAS REALIZADAS Y RESULTADOS OBTENIDOS

A fin de determinar la metodología de adquisición óptima que debe cumplimentar el sistema, se realizaron varias pruebas simuladas; observando en cada caso la calidad de los resultados obtenidos.

El criterio de evaluación se basó principalmente en dos factores: tiempo de procesamiento en la adquisición y generación de nube de puntos; y precisión en la triangulación de puntos. Dado que el hardware a utilizar se encuentra en función del método de adquisición, este punto fue una derivación directa de la determinación del primero.

Tal como se comentó previamente, las pruebas realizadas fueron basadas en simulaciones por computadora mediante la aplicación 3D Studio Max. Las mismas fueron de vital utilidad al momento de verificar configuraciones de hardware/software, ubicación de las cámaras y correctitud de los algoritmos.

La figura 4a presenta la renderización 3D de la nube de puntos obtenida mediante la primer técnica, correspondiente a la primer secuencia anteriormente mencionada.

La figura 4b presenta la renderización 3D de la nube de puntos obtenida mediante la segunda técnica, correspondiente a la segunda secuencia anteriormente mencionada. Cabe destacar que en esta interpretación debe contemplarse que la tetera se encuentra "abierta" y estirada (dada la forma en que Matlab realiza la renderización).

Como se muestra posteriormente, en la visualización en el entorno de renderización 3D será contemplado el ángulo de rotación de la captura para la reconstrucción del objeto de la figura 3.

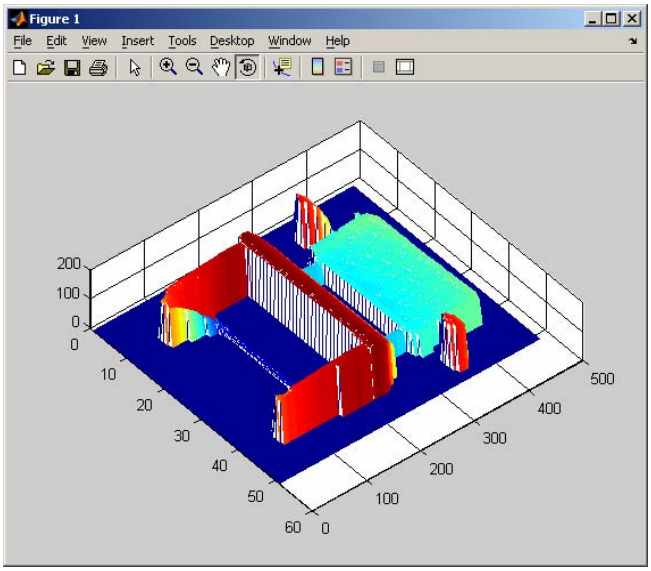


Figura 4a. Visualización de la matriz de profundidades en Matlab.

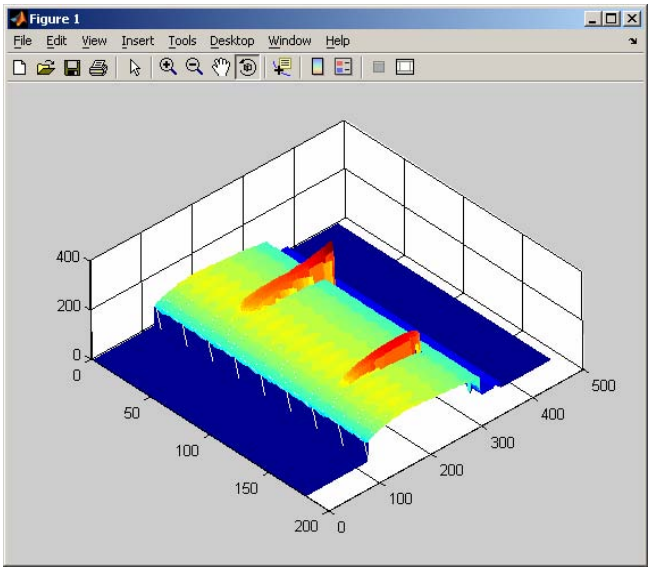


Figura 4b. Visualización de la matriz de profundidades en Matlab (rotación de objeto)

5 ENTORNO DE RENDERIZACIÓN 3D

El entorno de renderización 3D se encarga de procesar la nube de puntos almacenada en la matriz de profundidades y generar el objeto escaneado. Una vez obtenido éste, se realiza la visualización del mismo.

El entorno se encuentra en etapa de desarrollo. Sin embargo, la figura 5 presenta dos visualizaciones iniciales de las reconstrucciones realizadas. La figura 5a corresponde a la secuencia mostrada en la figura 2, mientras que la figura 5b corresponde a la secuencia de la figura 3.

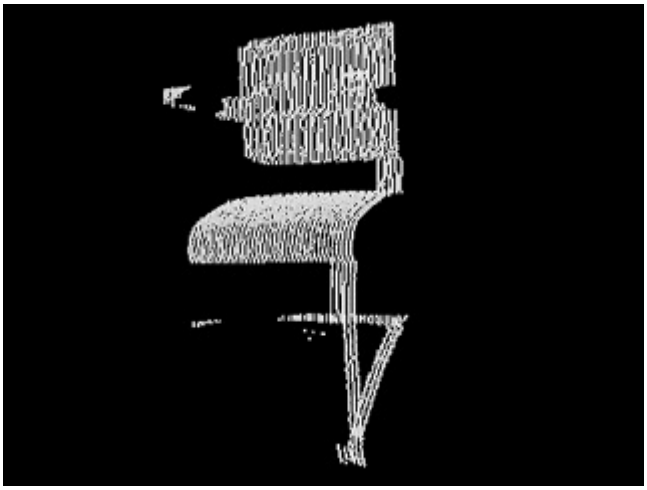


Figura 5a. Renderizaciones a partir de la matriz de profundidades

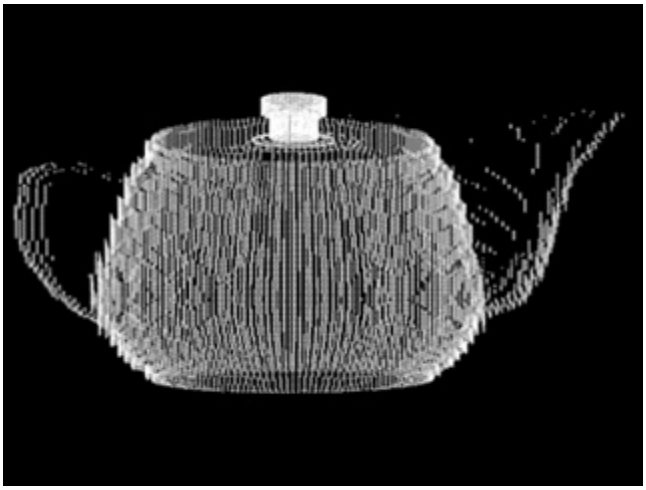


Figura 5b. Renderizaciones a partir de la matriz de profundidades (rotación del objeto)

El entorno permite observar la pieza desde cualquier posición que sea necesario, pudiendo rotar tanto la cámara como el objeto.

Aunque el sistema de adquisición se encuentra desarrollado utilizando la librería OpenCV [18], el desarrollo del entorno de visualización se está efectuando apoyándose en la API de Java 3D de Sun Microsystems [19]. Esto indica una independencia funcional de cada una de las partes que componen la implementación, permitiendo el desarrollo por separado de cada una de éstas.

6 IMPLEMENTACION DEL HARDWARE

Actualmente se encuentra en desarrollo el armazón para el sistema de captura, el cual contará con 2 cámaras, un láser, 2 motores y la estructura que lo soporta. La misma fue pensada para permitir un fácil armado y desarmado a fin de facilitar su portabilidad.

La estructura posee ciertas características que aumentan la versatilidad del sistema de captura, tales como regules de horizontalidad, posicionamiento de cámaras, láser y objeto en múltiples posiciones, etc.

La figura 6 muestra dos vistas del prototipo en el cual actualmente se está trabajando.

Figura 6. Hardware en desarrollo

7 CONCLUSIONES

Aunque todavía el proyecto no se encuentra finalizado, el estado actual del mismo permite prever resultados más que aceptables con la tecnología de hardware que se espera utilizar en conjunto con el software desarrollado.

Aunque las pruebas realizadas hasta el momento fueron efectuadas en un ambiente controlado, las mismas arrojan un balance positivo en cuanto a la relaciones de costo/beneficio y de precisión/performance.

Una vez finalizado el desarrollo del hardware se espera utilizar inicialmente el escáner para la digitalización 3D de vasijas y estatuillas pertenecientes a la colección del Museo de Ciencias Naturales de La Plata.

8 REFERENCIAS

[1] Surface Reconstruction by Integrating 3D and 2D Data of Multiple Views Maxime.
LHUILLIER, Long QUAN LASMEA UMR 6602 Department of Computer Scince Universit´e Blaise Pascal/CNRS, France Hong Kong University of Science and Technology

[2] Reality : an Interactive Reconstruction Tool of 3D Objects from Photographs.
S. Dedieu , P. Guitton, C. Schlick, and P. Reuter LaBRI (CNRS, Universit´e Bordeaux I, ENSEIRB)
351 cours de la Lib´eration, 33405 Talence Cedex, France.

[3] High-Resolution Structured Light Range Scanner with Automatic Calibration
ALEXANDER M. BRONSTEIN¤ MICHAEL M. BRONSTEIN z EYAL GORDON x RON KIMMEL

[4] Interactive 3-D Media with Structured Light Scanning
Nelson L. Chang - Imaging Systems Laboratory
HP Laboratories Palo Alto - HPL-2003-112 - May 29th , 2003

[5] Recuperación de Información Tridimensional usando Luz Estructurada
Venustiano Soancatl Aguilar. Tesis de Maestría en Ciencias Computacionales en Instituto Nacional de Astrofísica, Óptica y Electrónica.

[6] A Simple 3D Scanner
http://www.sjbaker.org/projects/scanner/

[7] Search Algorithms for Block-Matching in Motion Estimation. Deepak Turaga, Mohamed Alkanhal.
http://www.ece.cmu.edu/~ee899/project/deepak_mid.htm

[8] Optical Flow. David Marshall. http://www.cs.cf.ac.uk/Dave/Vision_lecture/node45.html

[9] Diffractive Optical Elements (DOEs)
http://www.stockeryale.com/i/lasers/accessories/diff_gratings.htm#patterns

[10] Laser line generators and other structured light pattern projectors
http://www.stockeryale.com/i/lasers/lasiris_lasers.pdf

[11] Laser and Eye Safety
http://www.stockeryale.com/i/lasers/safety.htm

[12] TRUCCO, VERRI, Introductory Techniques for 3D computer vision, Prentice Hall 1998.

[13] XU, ZHANG, Epipolar Geometry in Stereo, Motion and Object Recognition, A Unified Approach, Kluwer Academic Publishers 1996.

[14] HARTLEY, ZISSERMAN, Multiple View Geometry in computer vision, Cambridge University Press 2000.

[15] A Flexible New Technique for Camera Calibration. Zhengyou Zhang. Technical Report MSR-TR-98-71 - 2002

[16] Camera Calibration Toolbox for Matlab.
www.vision.caltech.edu\bouguetj\calib_doc\index.html

[17] Seguimiento tridimensional usando dos cámaras. Pablo Barrera González, José María Cañas Plaza - 2004

[18] OpenCV - Open Source Computer Vision Library Community.
http://groups.yahoo.com/group/OpenCV

[19] The Java 3D™ API Specification. Sun Microsystems, Inc.

DWT based Digital Watermarking Fidelity and Robustness Evaluation*

Franco Del Colle † **and Juan Carlos Gómez**

Laboratory for System Dynamics and Signal Processing
FCEIA, Universidad Nacional de Rosario
Riobamba 245 bis, (2000) Rosario, Argentina
{delcolle, jcgomez}@fceia.unr.edu.ar

Abstract

An Image AdaptiveWatermarking method based on the Discrete Wavelet Transform is presented in this paper. The robustness and fidelity of the proposed method are evaluated and the method is compared to state-of-the-art watermarking techniques available in the literature. For the evaluation of watermark transparency, an image fidelity factor based on a perceptual distortion metric is introduced. On the other hand, a degradation factor is introduced for the evaluation of watermark robustness against JPEG compression and resizing. The new fidelity metric allows a perceptually aware objective quantification of image fidelity. The suitability of the proposed metric for the fidelity evaluation of still image watermarking is supported by simulation results.

Keywords: Digital Watermarking, Discrete Wavelet Transform, Perceptual Metrics.

*This work is partially supported by CONICET.
†Author to whom all correspondence should be addressed.

1 INTRODUCTION

In the last decade, an important research effort has been devoted to the development of techniques adressing the issue of digital data protection. Among them, Digital Watermarking has become the most efficient and widely used.

Digital Watermarking refers to techniques that are used to protect digital data by imperceptibly embedding information (the watermark) into the original data in such a way that always remains present. As pointed out in [1], a set of requirements should be met by any watermarking technique. The main requirements are *perceptual transparency, payload of the watermark and robustness*. Perceptual transparency refers to the property of the watermark of been imperceptible in the sense that humans can not distinguish the watermarked images from the original ones by simple inspection. Payload of the watermark refers to the amount of information stored in the watermark, which in general depends on the application. Finally, robustness refers to the capacity of the watermark to remain detectable after alterations due to processing techniques or intentional attacks.

Good overviews on the state of the art of classical watermarking techniques can be found in the recent textbooks [1] and [6], and in [7], [9], [11] and the references therein.

Several techniques have been proposed in the literature for the watermarking of still images. From a general point of view, embedding is achieved by first extracting a set of features from the image to be watermarked, and then modifying them according to the watermark content. Thus, two steps are required to define the embedding process: choice of the features to be modified, and definition of the embedding rule. Several solutions have been proposed, leading to different watermarking schemes. The different approaches can be classified taking into account different aspects. When the domain in which the watermark is being embedded is considered, a classification in spatial domain techniques and transform domain techniques can be made [7]. When the watermark adaptation to the particular image is considered, a classification in Image Adaptive Watermarking (IAW) methods ([2], [11], [12], [13]) and Image Independent Watermarking (IIW) methods ([5], [10]) can be done. In the IAW techniques the length, location and amplitude of the watermark is adapted to the image characteristics, while in the IIW techniques the length of the inserted watermark does not depend on the particular image. This paper will focus on Image Adaptive Discrete Wavelet Transform (IADWT) domain watermarking techniques since they have proved to yield better results regarding transparency and robustness.

Typically, the evaluation of the watermarking scheme performance is carried out by quantifying the perceptual transparency of the watermark and its robustness against several signal processing operations such as compression, scaling, cropping, etc. [8]. In this paper, a new criterion for watermark transparency evaluation is proposed based on perceptual distortion metrics. In addition, a novel watermarking scheme in the DWT domain is proposed as a modification of the one in [12], which will prove to have a better performance. Further, the robustness of the proposed method against JPEG-compression and re-scaling, is analyzed.

The rest of the paper is organized as follows. In section 2, the IADWT technique is briefly described. A slight variation of the IADWT method in [12] is also introduced in this section. In section 3, the perceptual metric used for the evaluation of the fidelity performance is described and a new *fidelity factor* is introduced also there. The robustness criterion to evaluate watermark detectability after attacks is described in section 4. Results on the comparison between the proposed method and the method in [12] during insertion and detection are presented in section 5. Finally, some concluding remarks are given in section 6.

2 IMAGE ADAPTIVE DWT WATERMARKING

Image adaptive watermarking methods make use of visual models in order to determine the maximum length and power of the watermark according to the image capacity to ”hide information” without being perceptible. This capacity is calculated by means of the so called Just Noticeable Differences (JND) thresholds, which measure the smallest difference between images which is perceptually detectable by the human eye. In the DWT domain, these thresholds allows to determine the location of the transform coefficients and the amount that they can variate without being noticeable in the spatial domain.

In the watermark embedding scheme in [12], the watermark is modulated by the JND, and the coefficients are marked whenever they are greater than the JND threshold, *i.e.*

$$\widehat{X}^w(u,v) = \begin{cases} \widehat{X}(u,v) + J(u,v)w(\ell) & \widehat{X}(u,v) > J(u,v) \\ \widehat{X}(u,v) & \text{othewise} \end{cases} \quad (1)$$

where $\widehat{X}(u,v)$ and $\widehat{X}^w(u,v)$ are the DWT coefficients of the original image and the watermarked image respectively, and $J(u,v)$ is the JND matrix at the u,v frequency in the DWT domain.

In this scheme, the watermark sequence $w(\ell)$ is generated from a zero mean, unit variance, normally distributed random sequence. In this way, the watermark sequence weighted by the JND thresholds has lower power than the maximum power that can be inserted without causing noticeable distortions in the image. Figure 1 schematically depicts the image adaptive watermarking embedding scheme, where $X(i,j)$ denotes the original image and $X^w(i,j)$, the watermarked image.

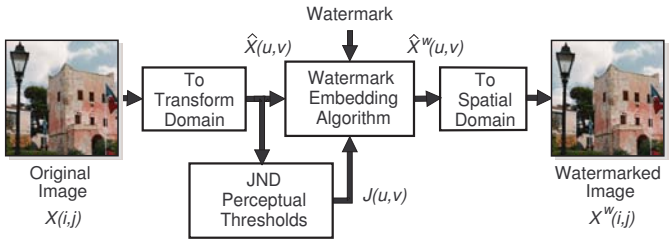


Figure 1: Image Adaptive Watermarking Embedding Scheme.

The JND thresholds are computed based on a perceptual model of the Human Visual System (HVS). A widely used perceptual model is the one introduced by Watson in [14]. This model takes into account frequency sensitivity, local luminance and contrast masking effects to determine an image-dependent quantization matrix, which provides the maximum possible quantization error in the DWT coefficients which is not perceptible by the HVS. This model has been used by the image compression standard JPEG2000, where the JND thresholds determine the optimal quantization step sizes or bit allocations for different parts of the image to be compressed.

In the watermark detection scheme the JND are calculated using the original image, then, the DWT coefficients of the original image are subtracted from the ones of the image suspected to be watermarked, and this difference is divided by the JND in order to obtain the received watermark. The correlation between the extracted watermark and the original one is then performed and the maximum value is determined, *i.e.*

$$w_e(\ell) = \frac{\widehat{X}^w(u,v) - \widehat{X}(u,v)}{J(u,v)} \quad \text{if } \widehat{X}(u,v) > J(u,v) \quad (2)$$

$$r_{w,w_e} = \frac{w_e(\ell) * w(-\ell)}{E_{w_e}.E_w} \tag{3}$$

where E_{w_e} and E_w are the energies of the extracted watermark sequence, $w_e(\ell)$, and the original watermark sequence, $w(\ell)$, respectively. Figure 2 schematically depicts the image adaptive watermarking detection scheme.

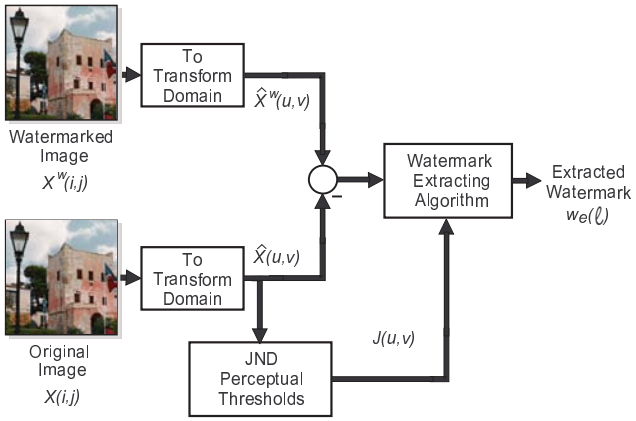


Figure 2: Image Adaptive Watermarking Detection Scheme.

The IADWT method has been studied in [12] and the authors pointed out two main advantages with respect to the IIW methods. First, non adaptive watermarking techniques are less robust, in order to guarantee transparency for a wide variety of input images. This is in contrast to the image adaptive approach which allows the watermark signal to reach the perceptual upper limit given by the JND thresholds. Second, for images with large uniform areas, heuristic techniques based on a global transform (like the one in [5]), could result in visible watermarks since the algorithms are not able to adapt to local image characteristics. On the other hand, the JND paradigm adapts the watermark not only to the global characteristics associated to the viewing conditions, but also to the local image characteristics associated with visual masking effects.

The following modification to the IADWT insertion scheme in (1) can be introduced

$$\widehat{X}^w(u,v) = \begin{cases} \widehat{X}(u,v) + J(u,v)w(\ell) & \widehat{X}(u,v) > J(u,v) > T \\ \widehat{X}(u,v) & \text{othewise} \end{cases} \tag{4}$$

This modified insertion scheme will be hereafter denoted as IADWT$_T$. The *rationale* for the constrain $J(u,v) > T$ is that when the JND thresholds are too small, the magnitude of the marking term in (4) becomes negligible. The introduction of the lower bound T has then the advantage of reducing the watermark length, improving in this way the fidelity and also the robustness, as will be illustrated in section 5.

The detection scheme in (2) has to be modified to take into account the modification in the insertion scheme, as follows

$$w_e(\ell) = \frac{\widehat{X}^w(u,v) - \widehat{X}(u,v)}{J(u,v)} \quad \text{if } \widehat{X}(u,v) > J(u,v) > T \tag{5}$$

3 FIDELITY EVALUATION USING PERCEPTUAL METRICS

In the evaluation of image watermarking methods it may be of interest to judge the fidelity of the inserted watermark. Basically the fidelity is a measure of the similarity between the images before and

after the insertion. For some watermarking applications, fidelity is the primary perceptual measure of concern, thus the watermarked image must be indistinguishable from the original.

In studies that involve the judgment by human beings, it is important to recognize that visual sensitivity can vary significantly from individual to individual, and moreover that sensitivity can change over time in any one individual. Therefore, it is common that studies involving human evaluation use a large number of subjects and perform a large number of trials, resulting in experiments that are statistical in nature and which become expensive if a large group is being considered. To avoid the dependence on human judgement it would be desirable to objectively quantify the fidelity of watermarked images based on a metric that takes into account the characteristics of the HVS.

Image fidelity metrics appeared in the context of imaging applications to quantify the distortion in images produced by image processing algorithms such as compression, halftoning, printing, etc. Different metrics have been proposed in the literature to measure image distortion (see [15] for a thorough treatment of distortion metrics and the more recent work [18]). Among them, the ones based on the characteristics of the HVS have proved to deliver the best results, since they take into account the different sensitivity of the human eye for color discrimination, contrast masking and texture masking.

A metric widely used to measure fidelity is the CIELAB metric [4] that specifies how to transform physical image measurements into perceptual differences (ΔE). The metric was derived from perceptual measurements of color discrimination of large uniform targets. A modification of the ΔE formula was released by CIE (International Commission on Illumination, Vienna) in 1994 based on new experimental data. The new formula was found to predict color differences slightly better than the old formula and it was named CIE94 [3].

An extension of CIELAB, named S-CIELAB [17], includes the spatial-color sensitivity of the human eye. The S-CIELAB metric incorporates the different spatial sensitivities of the three opponent color channels by adding a spatial pre-processing step before the standard CIELAB ΔE calculation. The S-CIELAB metric achieves this by removing the image components that cannot be seen by the naked eye. S-CIELAB consists of three processing steps. First, the original and distorted images, which are represented in a device-dependent space, are converted into a device-independent representation consisting of one luminance and two chrominance color components for each image, known as the YCbCr color space. Second, each component image is passed through a spatial filter that is selected according to the spatial sensitivity of the human eye for that color component. Third, the filtered images are transformed into the CIE-XYZ format such that the CIELAB color difference formula can be applied to give a S-CIELAB ΔE_{94} map, which indicates where the visible distortions are in the image, and how large the distortions are.

In [16] the authors test how well the S-CIELAB metric predicts image fidelity for a set of color images by comparison with two other metrics, namely, the widely used root mean square error (point-by-point RMS) computed in un-calibrated RGB values and the point-by-point CIELAB ΔE_{94} values.

Since the S-CIELAB metric takes into account the perceptual characteristics of the HVS, such as color discrimination, different spatial sensitivity, etc., this metric represents a natural choice for the quantification, in an objective way, of the fidelity of the watermarked image. To the best of the authors' knowledge, this perceptual evaluation of the fidelity has not been considered before in the context of Color Images Digital Watermarking.

To illustrate the use of the S-CIELAB metric, a region of the left image in Figure 3, delimited by the white square in the center image, is corrupted with zero mean unit variance additive Gaussian white noise. The right image shows the image distortion map corresponding to the noise corrupted image, where the S-CIELAB ΔE_{94} values are shown with a grayscale color map. The pixels where the S-CIELAB ΔE_{94} values are above a specified threshold are then marked in green. For reference

Figure 3: Left: Original Image. Center: Noisy Image. Right: Distortion Map.

purposes the edges of the original image are displayed in white. Note the reader that there are no perceptible differences between the original and corrupted images (left and center images in Figure 3, respectively).

The idea in this paper is to use distortion maps to compare watermarked image fidelity for the two insertion methods described in section 2. Due to the spatial distribution of the S-CIELAB ΔE_{94} errors in the distortion maps (the green marks in the right image of Figure 3) it is difficult to make a comparison of the different methods. To provide a unique parameter quantifying this fidelity, a pooling of the S-CIELAB ΔE_{94} errors is proposed as follows:

$$\mathcal{F} \triangleq \left(1 - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(S\Delta E_{94}(i,j)Mask(i,j))}{\sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{X_L(i,j)^2+X_a(i,j)^2+X_b(i,j)^2}}\right)\times 100 \tag{6}$$

where $S\Delta E_{94}$ is a matrix with the values of the S-CIELAB ΔE_{94} errors for each pixel, *i.e.* the image distortion map, *Mask* is a mask with ones in the positions where the S-CIELAB ΔE_{94} errors are above the threshold and zeros otherwise, X_L, X_a and X_b are the image components in the *Lab* color space. Values of $\mathcal{F}$ close to 100 % indicates that no perceptible distortion is present in the watermarked image.

The performance of the proposed metric will be compared in section 5 with that of a standard non perceptual metric based on the Root Mena Square (RMS) error. This metric, namely RMS Fit (RMS_{FIT}), is obtained by making a pooling of the RMS errors, resulting in:

$$RMS_{FIT} \triangleq \left(1 - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{\Delta X_R(i,j)^2+\Delta X_G(i,j)^2+\Delta X_B(i,j)^2}}{\sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{X_R(i,j)^2+X_G(i,j)^2+X_B(i,j)^2}}\right)\times 100 \tag{7}$$

where the subindexes *R*, *G* and *B* denote the corresponding image components in the RGB color space.

4 ROBUSTNESS EVALUATION

Another important issue when evaluating image watermarking methods is the robustness, *i.e.*, the capacity of the watermark to survive standard image processing alterations, such as lossy compression, scaling, cropping, printing and scanning, etc..

In this paper, robustness of the watermark against JPEG compression and re-scaling is evaluated by computing a degradation coefficient, *D*, which quantifies the degradation in the watermark detectability caused by these image processing tasks. To perform the robustness test, the watermarked image is subjected to each one of the above mentioned attacks, and then the watermark is extracted following the procedure described in section 2. The normalized cross-correlation between the original and the extracted watermarks is then computed. The *detectability degradation coefficient* is then

defined as,

$$\mathcal{D} \triangleq (1 - r_{w,w_e}(0)) \times 100 \tag{8}$$

where $r_{w,w_e}(k)$ denotes the normalized correlation between the original watermark, $w(\ell)$, and the extracted watermark, $w_e(\ell)$.

5 RESULTS

In order to compare the performance of the proposed watermarking scheme IADWT$_T$ and the IADWT in [12], a set of (256×256) natural color images was used. To make the results independent of the particular set of natural images considered, the same tests were also performed on synthetic pattern images with large uniform areas (like Image 4 in Figure 4.D) and images with predominant high frequency regions (like Image 5 in Figure 4.E).

Due to space limitations the results corresponding to only five images are presented in this paper. The original images, called Image 1 to Image 5, are shown in Figure 4.

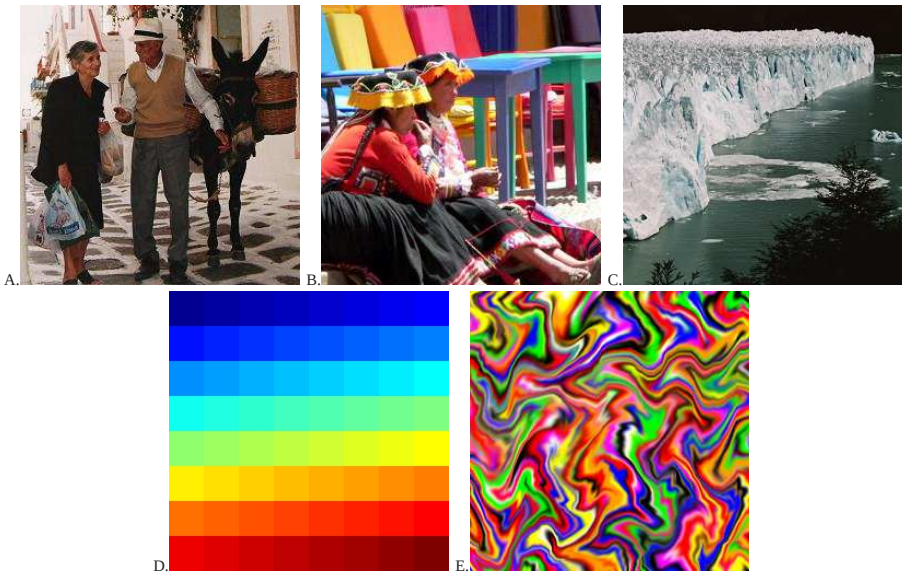


Figure 4: A. Image 1, B. Image 2, C. Image 3, D. Image 4 and E. Image 5.

5.1 Fidelity Evaluation results

In this section two separate tests to evaluate fidelity will be performed. The purpose of Test 1 in subsection 5.1.1 is to illustrate the fact that the fidelity factor $\mathcal{F}$ defined in (6) provides a much better assessment of image quality than the standard RMS_{FIT}. On the other hand, Test 2 in subsection 5.1.2 is designed to compare the fidelity of the two DWT based insertion schemes described in Section 2.

5.1.1 Fidelity Test 1

In order to illustrate the fact that the RMS_{FIT} does not provide an objective assessment of image quality, a watermarked image with a strong watermark was generated with the IIW embedding technique proposed in [5]. In this method, the watermark, denoted $\{w(\ell)\}_{\ell=1}^{L}$, is a length L sequence of

normally distributed, zero-mean unit-variance random numbers. Let $X(i,j)$ be the original image, $X^w(i,j)$ the watermarked image, and $\widehat{X}(u,v)$ and $\widehat{X}^w(u,v)$ their corresponding DCT coefficients. The embedding algorithm in [5] takes the L most significant non-DC DCT coefficients and marks them as follows:

$$\widehat{X}^w_\ell(u,v) = \widehat{X}_\ell(u,v)(1+\alpha w(\ell)) \tag{9}$$

where α is a scale factor which prevents unreasonable values for $\widehat{X}^w_\ell(u,v)$. The authors propose an empirically determined value of 0.1 for α and they choose to insert the watermark in the 1000 most significant non-DC DCT coefficients. After the watermark is embedded, the watermarked image $X^w(i,j)$ is obtained by inverse transforming all the DCT coefficients.

The original and the marked images are shown in the left and right sides of Figure 5, respectively. In this case the α parameter was chosen equal to 0.25, resulting in a fidelity factor $\mathcal{F} = 34.04\%$ and a $RMS_{FIT} = 91.26\%$. Based only on the RMS_{FIT} one would expect no noticeable distortions on the watermarked image which is not the case for this example (particularly in the sky portion at the top of the image). The fidelity factor $\mathcal{F}$ in turn gives a better assessment of image quality.

Figure 5: Left: Original Image. Right: Watermarked Image.

5.1.2 Fidelity Test 2

The values of the watermark length L, the normalized watermark energy in the spatial domain E (or equivalently, the normalized mean square error between the original and the watermarked images), the fidelity factor $\mathcal{F}$, and the RMS_{FIT} were computed for the five images in Figure 4, marked using the IADWT and IADWT$_T$ insertion schemes described in Section 2. The results are shown in Table 1.

As can be observed from the fifth column in Table 1 there is no noticeable difference between the fidelity, as measured by the RMS_{FIT}, using both insertion schemes. The difference is more noticeable using the proposed fidelity factor, as can be observed from the values in the fourth column.

The values of the fidelity factor, $\mathcal{F}$, in Table 1 show that the IADWT$_T$ method consistently outperforms the IADWT method regarding fidelity. Even for the case of images with large uniform color regions, as the one in Figure 4.D, where the image adaptive methods are supposed to work poorly [12], the IADWT$_T$ method produces non perceptible watermarks. On the other hand, the IADWT method introduces visible distortions, as can be observed from Figure 6 (see for instance the spots in the green regions of the upper left image).

The left columns in Figures 6 and 7 show the watermarked images corresponding to Image 1 and Image 4 using the above mentioned watermarking schemes (namely IADWT and IADWT$_T$ from top

Table 1: Experimental results on Fidelity Evaluation for Images 1 to 5.

	L	E $(\times 10^{-3})$	$\mathcal{F}$ (%)	RMS_{FIT} (%)
	Image 1			
IADWT	8347	1.40	92.27	97.45
IADWT$_T$	874	0.38	98.37	99.20
	Image 2			
IADWT	9314	1.26	94.13	97.52
IADWT$_T$	1036	0.37	98.50	99.22
	Image 3			
IADWT	8196	1.76	92.59	97.11
IADWT$_T$	1117	0.65	98.03	98.90
	Image 4			
IADWT	3002	0.12	99.63	99.17
IADWT$_T$	1138	0.07	99.82	99.67
	Image 5			
IADWT	11336	1.06	95.52	97.56
IADWT$_T$	1458	0.33	98.63	99.05

to bottom). The right columns show the corresponding distortion maps obtained after applying the S-CIELAB ΔE_{94} metric to the watermarked images. As expected, the distortion is larger in the regions with high frequency components, which results in a less perceptible watermark due to the masking phenomenon of the HVS.

5.2 Robustness Evaluation Results

In this subsection the robustness of the watermarked images against JPEG compression and re-scaling is evaluated, for both image adaptive DWT-based watermarking schemes.

5.2.1 JPEG Compression

The detectability degradation coefficient $\mathcal{D}$, as defined in (8), is computed for both image adaptive DWT-based watermarking schemes when JPEG-compression with quality factors in the range [95%-75%] is applied. The results for Images 1, 4 and 5 are shown in Figure 8 from left to right respectively. As can be observed the IADWT$_T$ watermarking scheme consistently outperforms the IADWT one regarding robustness against this image processing operation.

5.2.2 Re-scaling

The robustness against re-scaling is tested by first resizing the watermarked image to half of its size and then enlarging the image to its original size. Both image resizing operations are performed using the nearest neighbor interpolation method. The detectability degradation coefficient $\mathcal{D}$ is then computed for both image adaptive DWT-based watermarking schemes. The results are shown in Table 2. It can be observed that the IADWT$_T$ scheme outperforms the IADWT one for most of the

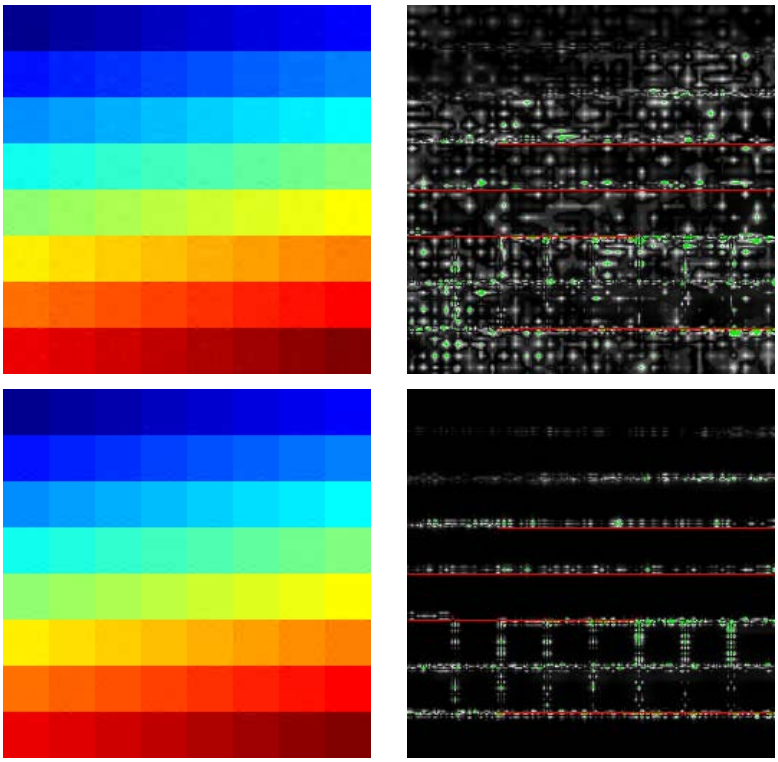


Figure 6: Left Column: Watermarked Image 4 using IADWT (top) and IADWT$_T$ (bottom). Right Column: Corresponding distortion maps.

images, the exception being for Image 4 which has large uniform color regions. Results not shown in Table 2 suggest that this behavior applies for images with large uniform color regions in general.

Table 2: Detectability degradation coefficient for 50% re-scaling.

	Scaling (50%)				
	Image 1	Image 2	Image 3	Image 4	Image 5
IADWT	89.11	89.13	89.70	29.37	85.63
IADWT$_T$	54.63	62.92	63.28	42.95	51.25

6 CONCLUDING REMARKS

An image *fidelity factor* based on the S-CIELAB ΔE_{94} perceptual distortion metric has been introduced in this paper for the purposes of evaluating the distortion introduced by different IADWT watermark insertion algorithms. The use of this metric allows a perceptually aware objective quantification of image fidelity. Simulation results show the suitability of the proposed metric in the framework of still image digital watermarking. In addition, a new IADWT watermarking scheme has been introduced. The robustness against compression and re-scaling, and the fidelity of the proposed method have been investigated and the results show that the proposed technique outperforms other methods available in the literature.

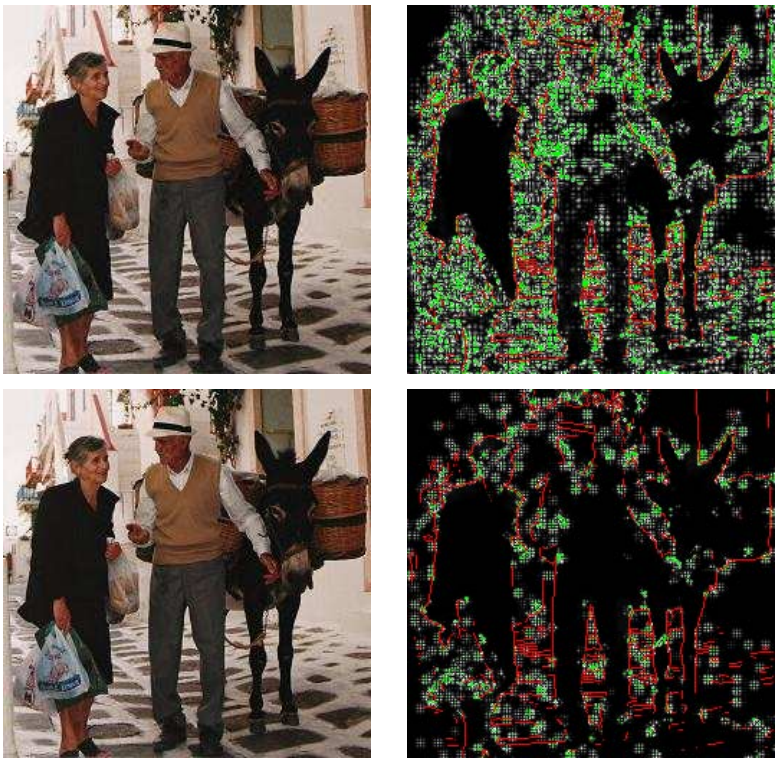


Figure 7: Left Column: Watermarked Image 1 using IADWT (top) and IADWT$_T$ (bottom). Right Column: Corresponding distortion maps.

REFERENCES

[1] M. Barni and F. Bartolini. *Watermarking Systems Engineering - Enabling Digital Assets and Other Applications*. Marcel Dekker, Inc., New York, 2004.

[2] M. Barni, F. Bartolini, and A. Piva. Improved wavelet-based watermarking through pixel-wise masking. *IEEE Transactions on Image Processing*, 10(5):783–791, May 2001.

[3] CIE: Int. Comm. on Illumination. Industrial colour-difference evaluation. Technical Report CIE 116-95, Austria, 1995.

[4] CIE: International Commission on Illumination. Recommendations on uniform color spaces, color difference equations, psychometrics color terms. Technical Report CIE 15 (E.-1.3.1), Supplement No.2, Bureau Central de la CIE, Vienna, Austria, 1971.

[5] I. Cox, J. Kilian, F. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, December 1997.

[6] I. Cox, M. Miller, and J.Bloom. *Digital Watermarking*. Morgan Kaufmann, San Francisco, 2002.

[7] G. Langelaar, I. Setyawan, and R.Lagendijk. Watermarking digital image and video data. *IEEE Signal Processing Magazine*, 17(5):20–46, 2000.

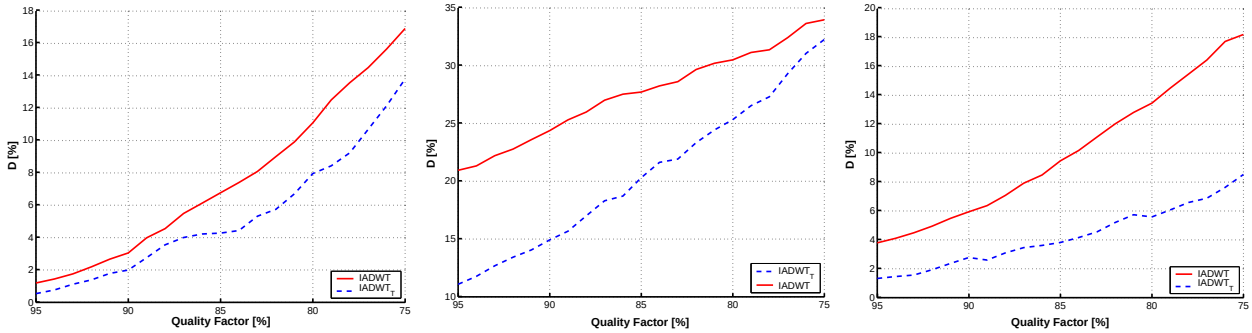


Figure 8: From Left to Right: Detectability degradation coefficient vs. JPEG Quality Factor for Images 1, 4 and 5, for IADWT and IADWT$_T$ watermarking schemes.

[8] E. Marini, F. Autrusseau, P. Le Callet, and P. Campisi. Evaluation of standard watermarking techniques. In E. J. Delp III and P. W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents IX, Proceedings of SPIE-IS& Electronic Imaging*, volume 6505, pages O1–O10, San Jose, CA, USA, 2007.

[9] F. Petitcolas. Watermarking schemes evaluation. *IEEE Signal Processing Magazine*, 17(5):58–64, September 2000.

[10] A. Piva, M. Barni, E Bartolini, and V. Cappellini. DCT-based watermark recovering without resorting to the uncorrupted original image. In *Proceedings of the International Conference on Image Processing*, volume 1, pages 520–523, 1997.

[11] C. Podilchuk and E. Delp. Digital watermarking: Algorithms and applications. *IEEE Signal Processing Magazine*, 18(4):33–46, July 2001.

[12] C. Podilchuk and W. Zeng. Image-adaptive watermarking using visual models. *IEEE Journal on Selected Areas in Communications*, 16(4):525–539, May 1998.

[13] M. Swanson, B. Zhu, and A. Tewfilc. Transparent robust image watermarking. In *Proceedings International Conference on Image Processing*, volume 3, pages 211–214, 1996.

[14] A. Watson, G. Yang, J. Solomon, and J. Villasenor. Visibility of wavelet quantization noise. 6(8):1164–1175, August 1997.

[15] S. Winkler. *Digital Video Quality Vision Models and Metrics*. John Wiley & Sons Ltd, Chichester, UK, 2005.

[16] X. Zhang and B. Wandell. Color image fidelity metrics evaluated using image distortion maps. *Signal Processing*, 70:201–214, 1998.

[17] Z. Zhang. A spatial extension to CIELAB for digital color image reproduction. *Society for Information Display Symposium Technical Digest*, 27:731–734, 1996.

[18] Z. Zhang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transaction on Image Processing*, 13(4):600–612, April 2004.

Matrices de subdivisión para curvas Beta-spline cúbicas

Liliana Castro
Dto. de Matemática, Univ. Nac. del Sur
Bahía Blanca, B8000CPB, Argentina
lcastro@uns.edu.ar

Silvia Castro
Dto. de Cs. e Ing. de la Computación, Univ. Nac. del Sur
Bahía Blanca, B8000CPB, Argentina
smc@cs.uns.edu.ar

Susana Kahnert
Dto. de Cs. e Ing. de la Computación, Univ. Nac. del Sur
Bahía Blanca, B8000CPB, Argentina
sak@cs.uns.edu.ar

Diana Salgado
Dto. de Matemática, Univ. Nac. del Sur
Bahía Blanca, B8000CPB, Argentina
dsalgado@uns.edu.ar

Abstract

In this paper we use a subdivision technique to calculate the control points that subdivide polinomial curves. If $\mathbf{P} = (P_0, ..., P_n)$ is the control polygon and $\mathbf{B}[\mathbf{P}]$ is the polinomial curve of degree n for which a subdivision algorithm is to be constructed, we use matrix operations to obtain the left polygon $\mathbf{L}$ and the right polygon $\mathbf{R}$ that aproximate the curve $\mathbf{B}[\mathbf{P}]$. Each one of the polygons $\mathbf{P}$, $\mathbf{L}$, and $\mathbf{R}$ represent a set of points in the plane. In this work we have considered the case of cubic Beta-spline curves, with parameters β_1 and β_2, and we obtained the subdivision curves for different values of these parameters. We explicitely detail the subdivision matrices that we have used for each case, and present the graphic representation of the subpolygons obtained in the different steps of the subdivision.

Resumen

En este trabajo se emplea una técnica de subdivisión para calcular los puntos de control que subdividen a las curvas polinómicas.

Sea $\mathbf{P} = (P_0, ..., P_n)$ el polígono de control y $\mathbf{B}[\mathbf{P}]$ la curva polinómica de grado n para la cual se construye el algoritmo de subdivisión. Mediante operaciones matriciales se obtienen $\mathbf{L}$

y **R**, polígonos a izquierda y a derecha, respectivamente, que aproximan a la curva **B**[**P**]. Cada uno de los polígonos **P**, **L**, y **R** representan un conjunto de puntos en el plano.

Consideramos el caso de las curvas Beta-spline cúbicas, con parámetros β_1 y β_2, y realizamos la subdivisión para distintos valores de estos parámetros. Detallamos explícitamente las matrices de subdivisión utilizadas para cada caso, así como también la representación gráfica de los subpolígonos obtenidos en los distintos pasos de la subdivisión.

Palabras claves

matrices de subdivisión, polígono de control, curvas Beta-spline

1. INTRODUCCION

Recordemos que un algoritmo de subdivisión es una técnica para encontrar polígonos de control **L** (a izquierda) y **R** (a derecha) tales que dividen a la curva $\mathbf{B}[\mathbf{P}](t)$ en algún valor $c \in (0,1)$. Si suponemos que $\mathbf{B}[\mathbf{P}]([a,b])$ representa la curva cuando el parámetro varía sobre el intervalo $[a,b]$, entonces

$$\mathbf{B}[\mathbf{P}]([0,1]) = \mathbf{B}[\mathbf{L}]([0,1]) \cup \mathbf{B}[\mathbf{R}]([0,1]).$$

Más precisamente,

$$\mathbf{B}[\mathbf{L}]([0,1]) = \mathbf{B}[\mathbf{P}]([0,c])$$
$$\mathbf{B}[\mathbf{R}]([0,1]) = \mathbf{B}[\mathbf{P}]([c,1]).$$

Goldman Ronald y DeRose Tony [4] desarrollaron una forma de encontrar las matrices de subdivisión a izquierda y a derecha que se aplican a un determinado polígono de control para obtener subpolígonos a izquierda y a derecha, considerando para ello el caso de la subdivisión de curvas polinómicas.

Nuestro interés es mostrar explícitamente las matrices de subdivisión para curvas Beta-splines cúbicas, para distintos valores de los parámetros de sesgo y de tensión. Estas matrices nos brindarán los puntos de control correspondientes a **L** y **R**.

Asimismo queremos mostrar, gráficamente, la influencia de estos parámetros en la subdivisión.

2. MATRICES DE SUBDIVISION

En esta sección se desarrolla una técnica para calcular los puntos de control que subdividen a las curvas o superficies polinómicas.

Sea $B(t) = (B_0(t), ..., B_n(t))$, con $B_0(t), ..., B_n(t)$ funciones que forman una base para la curva polinómica $\mathbf{B}[\mathbf{P}](t)$ de grado n, para la cual será construido un algoritmo de subdivisión, siendo $\mathbf{P} = (P_0, ..., P_n)$ el polígono de control y $P_0, ..., P_n$ los puntos de control. Es decir, consideremos

$$\mathbf{B}[\mathbf{P}](t) = \sum_{k=0}^{n} B_k(t) P_k = B(t) * \mathbf{P}^t,$$

siendo $\mathbf{P}^t$ la traspuesta de $\mathbf{P}$ y " $*$ " representa la multiplicación de matrices. Las funciones $B_0(t), ..., B_n(t)$ son continuas, suman uno y, dado que forman una base, son linealmente independientes sobre el intervalo [0,1]. Estas tres propiedades nos aseguran que, si un algoritmo de subdivisión es aplicado a la curva $\mathbf{B}[\mathbf{P}](t)$, entonces los polígonos de control convergen a la curva [4].

Como $B(t)$ es una base para el espacio de los polinomios de grado n, existen matrices S_L y S_R [4] que dependen de c, $c \in (0,1)$ tales que:

$$\begin{cases} B(ct) = B(t) * S_L \\ B(c + (1-c)t) = B(t) * S_R, \end{cases} \tag{1}$$

donde S_L y S_R son matrices que reparametrizan las funciones $B_0(t), ..., B_n(t)$. En el caso de la subdivisión en el punto medio, consideramos $c = \frac{1}{2}$.

Sea $\mathbf{P} = (P_0, ..., P_n)$ el polígono de control; buscamos expresiones para $\mathbf{L}$ y $\mathbf{R}$ tales que:

$$\begin{cases} \mathbf{B}[\mathbf{L}](t) = \mathbf{B}[\mathbf{P}](ct) \\ \mathbf{B}[\mathbf{R}](t) = \mathbf{B}[\mathbf{P}](c + (1-c)t), \end{cases} \tag{2}$$

donde $\mathbf{L}$ y $\mathbf{R}$ son los subpolígonos a izquierda y a derecha respectivamente, en los cuales queda dividido el polígono original $\mathbf{P}$.

En forma matricial, esto se puede escribir de la siguiente manera:

$$B(t) * \mathbf{L}^t = B(ct) * \mathbf{P}^t$$

$$(B_0(t) \quad \cdots \quad B_n(t)) * \begin{pmatrix} L_0 \\ \vdots \\ L_n \end{pmatrix} = (B_0(ct) \quad \cdots \quad B_n(ct)) * \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix}.$$

Sustituyendo por la primera ecuación de (1) obtenemos que

$$B(t) * \mathbf{L}^t = B(t) * S_L * \mathbf{P}^t.$$

Como $B_0(t), ..., B_n(t)$ son linealmente independientes, se concluye que

$$\mathbf{L}^t = S_L * \mathbf{P}^t \tag{3}$$

$$\begin{pmatrix} L_0 \\ \vdots \\ L_n \end{pmatrix} = S_L * \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix}.$$

Similarmente obtenemos que:

$$\mathbf{R}^t = S_R * \mathbf{P}^t.$$

Por lo tanto, las matrices que reparametrizan las funciones de la combinación lineal sirven, además, para subdividir los polígonos de control. El próximo lema nos da expresiones explícitas para las matrices.

Lema 1 *Sea $T = (t_0, ..., t_n)$ una sucesión de constantes tales que $0 \leq t_k \leq 1$ (ver lema 2.1 de [4]), y sea $0 \leq c \leq 1$ una constante. Entonces S_L y S_R están dadas explícitamente por*

$$\begin{cases} S_L = B(T)^{-1} * B(cT) \\ S_R = B(T)^{-1} * B(c + (1-c)T) \end{cases} \tag{4}$$

donde $cT = (ct_0, ..., ct_n)$, $c + (1-c)T = (c + (1-c)t_0, ..., c + (1-c)t_n)$ *y*

$$B(T) = \begin{pmatrix} B_0(t_0) & \cdots & B_n(t_0) \\ \vdots & & \vdots \\ B_0(t_n) & \cdots & B_n(t_n) \end{pmatrix}$$

Demostración: De (1), $B(ct_i) = B(t_i) * S_L$, con $i = 0, ..., n$, y ésto en forma matricial es $B(cT) = B(t) * S_L$

$$\begin{pmatrix} B_0(ct_0) & B_1(ct_0) & \cdots & B_n(ct_0) \\ B_0(ct_1) & B_1(ct_1) & \cdots & B_n(ct_1) \\ \vdots & \vdots & & \vdots \\ B_0(ct_n) & B_1(ct_n) & \cdots & B_n(ct_n) \end{pmatrix} = \begin{pmatrix} B_0(t_0) & B_1(t_0) & \cdots & B_n(t_0) \\ B_0(t_1) & B_1(t_1) & \cdots & B_n(t_1) \\ \vdots & \vdots & & \vdots \\ B_0(t_n) & B_1(t_n) & \cdots & B_n(t_n) \end{pmatrix} * S_L$$

Así, $S_L = B(T)^{-1} * B(cT)$.
En forma similar se encuentra la expresión de S_R.□

Estas matrices de subdivisión pueden determinarse a partir de las matrices de subdivisión de Bézier S_L^B y S_R^B, como en el caso de cualquier técnica polinomial. Dadas las funciones de Bézier de grado n (los polinomios de Bernstein) denotadas por $b_0^n(t), ..., b_n^n(t)$, es decir

$$b_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, i = 0, ..., n,$$

las matrices S_L^B y S_R^B se obtienen a partir de un algoritmo desarrollado en [2] y en [3], y están dadas por

$$S_L^B = \begin{pmatrix} b_0^0(c) & 0 & 0 & \cdots & 0 \\ b_0^1(c) & b_1^1(c) & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ b_0^n(c) & b_1^n(c) & b_2^n(c) & \cdots & b_n^n(c) \end{pmatrix} \tag{5}$$

$$S_R^B = \begin{pmatrix} b_0^n(c) & b_1^n(c) & \cdots & b_n^n(c) \\ 0 & b_0^{n-1}(c) & \cdots & b_{n-1}^{n-1}(c) \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & b_0^0(c) \end{pmatrix}. \tag{6}$$

Sea C la matriz cuyas componentes son los coeficientes de las combinaciones lineales que surgen de expresar cada función de la base $B(t) = (B_0(t), ..., B_n(t))$ en términos de las funciones de la base de Bernstein $b(t) = (b_0^n(t), ..., b_n^n(t))$, esto es C es tal que:

$$\begin{aligned} B(t) &= b(t) * C \\ b(t) &= B(t) * C^{-1}. \end{aligned}$$

Entonces por (1),

$$B(t) * S_L = B(ct) = b(ct) * C = b(t) * S_L^B * C = B(t) * C^{-1} * S_L^B * C.$$

Por la linealidad de $B(t)$ se verifica que

$$S_L = C^{-1} * S_L^B * C. \tag{7}$$

En forma similar, se demuestra que

$$S_R = C^{-1} * S_R^B * C. \tag{8}$$

De (7) y (8) resulta que S_L es similar a S_L^B y que S_R es similar a S_R^B.

Las matrices de los ejemplos que daremos en este trabajo fueron construidas usando las ecuaciones (7) y (8).

Concluimos esta sección con algunos resultados que nos permiten asegurar con precisión cuándo los polígonos de subdivisión permanecen en la cápsula convexa del polígono de control original.

Lema 2 *Sea* $T = (t_0, ..., t_n)$ *una sucesión de números reales tales que* $0 \leq t_k \leq 1$. *Consideremos*

$$B(T) = \begin{pmatrix} B_0(t_0) & \cdots & B_n(t_0) \\ \vdots & & \vdots \\ B_0(t_n) & \cdots & B_n(t_n) \end{pmatrix}$$

y sea $B(T)^{-1} = \{b_{ij}(T)\}$. *Entonces* $\sum_j b_{ij}(T) = 1, i = 0, ..., n$. *Esto es, las filas de* $B(T)^{-1}$ *suman uno.*

Demostración: Sea $\mathbf{u} = (1, ..., 1)$ un vector fila de n componentes. Sabemos que $\sum_{j=0}^{n} B_j(t_i) = 1, i = 0, ..., n$, o lo que es equivalente

$$B(T) * \mathbf{u}^t = \mathbf{u}^t.$$

Multiplicamos ambos miembros por $B(T)^{-1}$, obtenemos

$$B(T)^{-1} * \mathbf{u}^t = \mathbf{u}^t,$$

es decir

$$\sum_j b_{ij}(T) = 1, i = 0, ..., n. \square$$

Lema 3 *Las filas de* $S_L = (l_{ij})$ *suman uno. Esto es* $\sum_j l_{ij} = 1$. *Análogamente, las filas de* $S_R = (r_{ij})$ *suman uno.*

Demostración: Recordemos que $B_0(t), ..., B_n(t)$ suman uno $\forall t \in [0, 1]$ y como cada elemento de cT está en [0,1], las filas de $B(cT)$ deben sumar uno. Esto es, si $\mathbf{u} = (1, 1, ..., 1)$, entonces

$$B(cT) * \mathbf{u}^t = \mathbf{u}^t.$$

Multiplicando a ambos lados por $B(T)^{-1}$ y aplicando el Lema 2 y el Lema 1, obtenemos

$$S_L * \mathbf{u}^t = B(T)^{-1} * B(cT) * \mathbf{u}^t = B(T)^{-1} * \mathbf{u}^t = \mathbf{u}^t.$$

Entonces las filas de S_L suman uno. Un análisis similar muestra que las filas de S_R suman uno.$\square$

Lema 4 *Se garantiza que los polígonos de subdivisión* $\boldsymbol{L}$ *(*$\boldsymbol{R}$*) permanecen por completo en la cápsula convexa de* $\boldsymbol{P}$ *si y sólo si* S_L *(*S_R*) consisten solamente de elementos no negativos.*

Demostración: Sea $\mathbf{L} = (L_0, ..., L_n)$, $S_L = (l_{ij})$. Entonces, de (3) sigue que

$$L_i = \sum_j l_{ij} P_j.$$

Así, L_i permanece en la cápsula convexa de P_j si y sólo si $l_{i0}, ..., l_{in}$ son todos no negativos y

$$\sum_j l_{ij} = 1. \tag{9}$$

Esto es, las filas de S_L deben sumar uno y sus entradas deben ser no negativas. Pero, por el Lema 3, se sabe que las filas suman uno, entonces, todo lo que se requiere es que las entradas sean no negativas.

Un argumento similar se usa para probar el resultado para los puntos de control de $\mathbf{R}$.□

3. EJEMPLO: CURVA BETA-SPLINE CUBICA

La Beta-spline cúbica es una generalización de la B-spline cúbica uniforme, en la cual la propiedad de continuidad paramétrica de segundo grado (C^2) es reemplazada por continuidad geométrica (G^2). A continuación repasaremos estos dos conceptos.

Dada una curva paramétrica cúbica a trozos de ecuación $Q(t)$, con $t \in [t_0, t_m]$, el *vector tangente unitario* a $Q(t)$ está definido por :

$$T(t) = \frac{Q^{(1)}(t)}{|Q^{(1)}(t)|};$$

y el *vector curvatura* por:

$$K(t) = k(t)N(t) = k(t)\frac{T^{(1)}(t)}{|T^{(1)}(t)|};$$

donde $k(t)$ es la curvatura de $Q(t)$ en t y $N(t)$ es el vector normal unitario. Los vectores $T(t)$ y $K(t)$ representan la dirección del movimiento y la curvatura en un punto de la curva.

Notaremos $Q^{(n)}(t_i^+)$ y $Q^{(n)}(t_i^-)$, $n = 0, 1, 2$, los límites a derecha y a izquierda de $Q^{(n)}$ en t_i, respectivamente, siendo $Q^{(n)}$ la derivada n-ésima de $Q(t)$. Si deseamos que $Q(t)$, $T(t)$ y $K(t)$ sean continuas en algún t_i, es decir, que la curva tenga *continuidad geométrica de segundo grado* (indica que sea G^2) , se deben verificar las siguientes condiciones para cada t_i y para cada $\beta 1_i$ y $\beta 2_i$:

i) $Q(t_i^-) = Q(t_i^+)$

ii) $\beta 1_i Q^{(1)}(t_i^-) = Q^{(1)}(t_i^+)$

iii) $\beta 1_i^2 Q^{(2)}(t_i^-) + \beta 2_i Q^{(1)}(t_i^-) = Q^{(2)}(t_i^+)$.

La primera condición requiere la continuidad del vector posición. La segunda, se refiere a que las derivadas laterales primeras con respecto al parámetro en t_i sean colineales, pero admite que difieran en una constante; esto significa que puede existir un cambio en la velocidad pero no en la dirección. La última condición, está relacionada con la continuidad del vector curvatura.

Estas condiciones son menos restrictivas que las que verifican las curvas B-splines cúbicas uniformes ($\beta 1_i = 1$ y $\beta 2_i = 0$), ya que éstas tienen *continuidad paramétrica de segundo grado*, es decir, verifican:

i) $Q(t_i^-) = Q(t_i^+)$

ii) $Q^{(1)}(t_i^-) = Q^{(1)}(t_i^+)$

iii) $Q^{(2)}(t_i^-) = Q^{(2)}(t_i^+)$.

En el ejemplo que estamos considerando, un trozo de la curva Beta-spline cúbica está caracterizado por las funciones $B_0(t), ..., B_3(t)$ definidas por:

$$\begin{cases} B_0(t) = \frac{1}{\gamma}[2\beta_1^3 - 6\beta_1^3 t + 6\beta_1^3 t^2 - 2\beta_1^3 t^3] \\ B_1(t) = \frac{1}{\gamma}[(\beta_2 + 4\beta_1^2 + 4\beta_1) + (6\beta_1^3 - 6\beta_1)t - (3\beta_2 + 6\beta_1^3 + 6\beta_1^2)t^2 \\ \qquad +(2\beta_2 + 2\beta_1^3 + 2\beta_1^2 + 2\beta_1)t^3] \\ B_2(t) = \frac{1}{\gamma}[2 + 6\beta_1 t + (3\beta_2 + 6\beta_1^2 - (2\beta_2 + 2\beta_1^2 + 2\beta_1 + 2)t^3)] \\ B_3(t) = \frac{1}{\gamma}2t^3, \end{cases}$$

donde $\gamma = \frac{1}{\beta_2+12}$, β_1 es el parámetro de sesgo y β_2 es el parámetro de tensión [1].

Si se desea que la curva posea la propiedad de la cápsula convexa, basta con asignarle valores a los parámetros, de tal manera que las funciones $B_0(t), ..., B_3(t)$ sean no negativas.

La matriz de subdivisión $S_L = [l_{ij}]$ se construye a partir de la ecuación (7) $S_L = C^{-1} * S_L^B * C$. La matriz $S_R = [r_{ij}]$ es tal que $r_{ij} = l_{5-i,5-j}$ con $i, j = 1, ..., 4$.

Para construir la matriz C realizamos el siguiente procedimiento:

- Expresamos cada $B_i(t)$, $i = 0, ..., 3$, como una combinación lineal de las funciones $b_0^3(t)$, $b_1^3(t)$, $b_2^3(t)$ y $b_3^3(t)$.
- Cada uno de los coeficientes de esta combinación lineal formarán la $(i+1)$-ésima columna de C, con $i = 0, ..., 3$.

La matriz C presenta la siguiente forma general:

$$\begin{pmatrix} 2\gamma\beta_1^3 & \gamma(\beta_2 + 4\beta_1^2 + 4\beta_1) & 2\gamma & 0 \\ 0 & \gamma(2\beta_1^3 + 2\beta_1 + \beta_2 + 4\beta_1^2) & \gamma(2\beta_1 + 2) & 0 \\ 0 & \gamma(2\beta_1^3 + 2\beta_1^2) & \gamma(\beta_2 + 2\beta_1^2 + 4\beta_1 + 2) & 0 \\ 0 & 2\gamma\beta_1^3 & \gamma(\beta_2 + 4\beta_1 + 4\beta_1^2) & 2\gamma \end{pmatrix}$$

Debido a que estamos considerando $n = 3$, la matriz S_L^B será de orden cuatro, (ver (5)), y cuatro serán también los puntos de control del polígono original $\mathbf{P}$.

Una vez obtenidas las matrices C y S_L^B estamos en condiciones de hallar S_L. Luego, a partir de ésta, se calcula S_R. Si multiplicamos S_L y S_R por el vector columna $\mathbf{P}^t$, obtenemos los vectores columnas $\mathbf{L}^t$ y $\mathbf{R}^t$, respectivamente.

Utilizando el software Mathematica desarrollamos un algoritmo que nos permite hallar S_L y S_R para distintos valores de β_1 y β_2 y para un c cualquiera (ver Apéndice A).

En lo que respecta a la representación gráfica, hemos elaborado un algoritmo, utilizando el software MATLab, que genera los puntos de control a izquierda y a derecha, en los distintos pasos de la subdivisión y grafica los polígonos que estos puntos determinan, respectivamente.

En todos los ejemplos hemos considerado $\beta_1 = 1$, $c = 1/2$ y comenzamos la subdivisión a partir de los cuatro puntos de control originales y su polígono asociado.

En primer lugar, sea $\beta_2 = 0$ y $c = \frac{1}{2}$, es decir realizamos la subdivisión en el punto medio. Luego de realizar tres pasos en la subdivisión, la unión de las gráficas de los subpolígonos

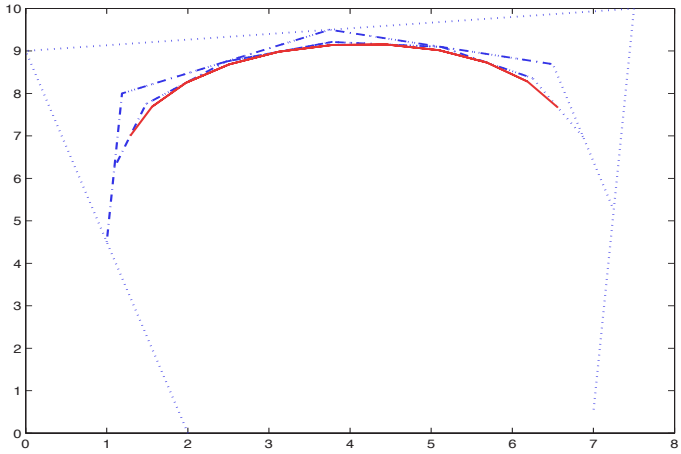


Figura 1: Tres pasos de la subdivisión en el punto medio para $\beta_1 = 1$ y $\beta_2 = 0$

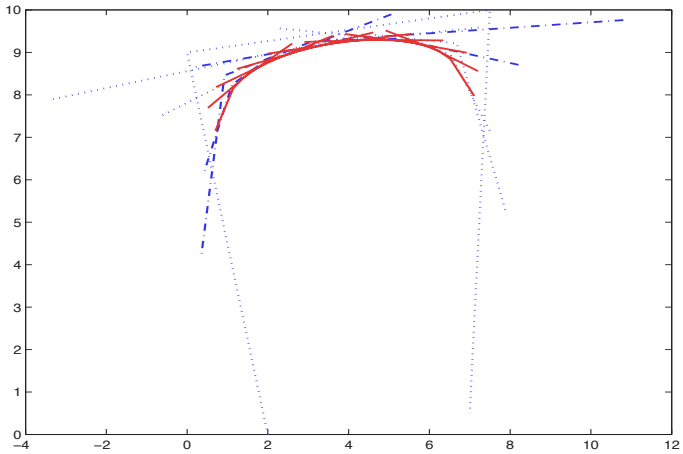


Figura 2: Tres pasos de la subdivisión en el punto medio para $\beta_1 = 1$ y $\beta_2 = 5$

obtenidos generan un polígono final que converge a la curva Beta-spline cúbica y que, además, está contenido en la cápsula convexa del polígono original, como puede observarse en la Figura 1.

Si suponemos $\beta_2 = 5$, la primera y la última fila de S_L contienen elementos negativos, por lo tanto, los primeros y los últimos puntos de los polígonos de subdivisión no quedan incluidos en la cápsula convexa del polígono original. Esto ocurre siempre que β_2 sea positivo.

El polígono formado por la unión de las gráficas de los polígonos de subdivisión **L** y **R** no describen una curva continua, como se puede observar en la Figura 2. Sin embargo, luego de sucesivos pasos de subdivisión, la unión de las gráficas de los polígonos de subdivisión converge a la curva original.

Por último, consideramos $\beta_2 = -10$, y observamos, en este caso, un comportamiento similar al caso anterior, (ver la Figura 3). Es decir, si $\beta_2 < 0$, cuanto menor es el valor de este parámetro, la curva no pertenece a la cápsula convexa.

4. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo hemos mostrado algunos ejemplos de subdivisión de curvas Beta-spline cúbicas y sus correspondientes matrices de subdivisión.

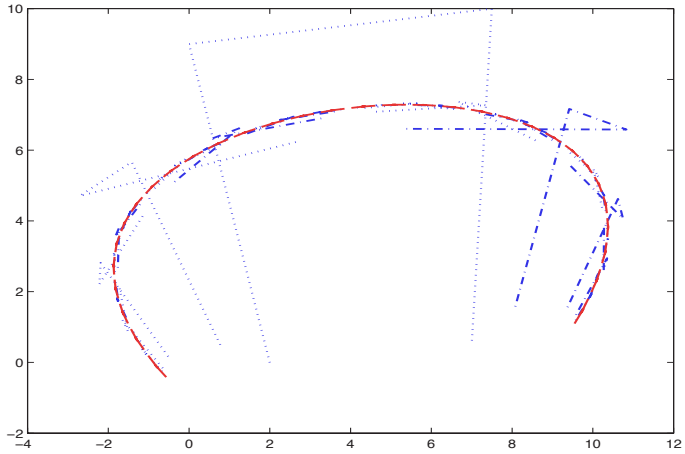


Figura 3: Cinco pasos de la subdivisión en el punto medio para $\beta_1 = 1$ y $\beta_2 = -10$

Se construyó un algoritmo que genera estas matrices para distintos valores de los parámetros de sesgo y tensión y para la subdivisión tanto en el punto medio como en otros casos.

En todos los ejemplos mostrados se trabajó con funciones de la base que son continuas, forman una partición de la unidad y son linealmente independientes, lo cual asegura la convergencia uniforme de los polígonos de subdivisión a la curva original. No se hizo uso de la propiedad de la cápsula convexa, ya que esta propiedad no es necesaria para la convergencia de un algoritmo de subdivisión [4].

Está claro que, a partir de cuatro puntos de control, se genera un solo trozo de la curva. Una extensión a un conjunto con mayor número de puntos de control forma parte de un trabajo futuro, así como la extensión al caso de la subdivisión de superficies Beta-spline cúbicas.

Referencias

[1] Barsky B.A. and Beatty J.C. Local control of bias and tension in Beta-splines. *ACM Transactions on Graphics*, 2(2):109–134, 1983.

[2] Goldman R.N. Using degenerate Bézier triangles and tetrahedra to subdivide Bézier curves. *Computer Aided Geometric Design 14*, 6:307–311.

[3] Goldman R.N. Markov chains and computer-aided geometric design, part II: Examples and subdivision matrices. *ACM Transactions on Graphics*, 4(1):12–40, 1985.

[4] Goldman R.N. and DeRose T.D. Recursive subdivision without the convex hull property. *Computer Aided Geometric Design 3*, 4:247–265, December 1986.

A. Apéndice

El programa para hallar las matrices de subdivisión a derecha y a izquierda fue realizado utilizando el software Mathematica.

```
b := Function[{i, n, t}, n! / ((n - i)! i!) t^i (1 - t)^(n-i)];

Matrices[β1_, β2_, t_] := Module[{SLB, SRB, C, SL, SR, γ},

  SLB = {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};

  For[i = 1, i < 5, For[j = 1, j < i + 1, SLB[[i, j]] = b[j - 1, i - 1, t]; j++]; i++];

  SRB = {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};

  For[i = 1, i < 5, For[j = i, j < 5, SRB[[i, j]] = b[j - i, 4 - i, t]; j++]; i++];

  γ = 1 / (12 + β2);

  C = {{2 γ β1^3, γ (β2 + 4 β1^2 + 4 β1), 2 γ, 0},
       {0, γ (2 β1^3 + 2 β1 + β2 + 4 β1^2), γ (2 β1 + 2), 0},
       {0, γ (2 β1^3 + 2 β1^2), γ (β2 + 2 β1^2 + 4 β1 + 2), 0},
       {0, 2 γ β1^3, γ (β2 + 4 β1 + 4 β1^2), 2 γ}};

  SL = Simplify[Inverse[C].SLB.C]; SR = {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};

For[i = 1, i < 5, For[j = 1, j < 5, SR[[i, j]] = SL[[5 - i, 5 - j]]; j++]; i++];
  {Print[MatrixForm[SL]], Print[MatrixForm[SR]]}]
```

A continuación mostramos cómo se obtienen las matrices correspondientes a los ejemplos que hemos desarrollado en este trabajo. En primer lugar, consideramos $\beta 1 = 1$, $\beta 2 = 0$ y $t = 1/2$ (la subdivisión se realiza en el punto medio), es decir calculamos Matrices[1,0,1/2] y obtenemos las matrices siguientes:

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \end{pmatrix} \quad \begin{pmatrix} \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

En el segundo ejemplo consideramos $\beta 1 = 1$, $\beta 2 = 5$ y $t = 1/2$, y luego de calcular Matrices[1,5,1/2] obtenemos:

$$\begin{pmatrix} \frac{79}{153} & \frac{39}{68} & -\frac{55}{612} & 0 \\ \frac{11}{153} & \frac{14}{17} & \frac{16}{153} & 0 \\ \frac{5}{306} & \frac{39}{68} & \frac{251}{612} & 0 \\ -\frac{65}{1224} & -\frac{29}{68} & \frac{829}{612} & \frac{1}{8} \end{pmatrix} \quad \begin{pmatrix} \frac{1}{8} & \frac{829}{612} & -\frac{29}{68} & -\frac{65}{1224} \\ 0 & \frac{251}{612} & \frac{39}{68} & \frac{5}{306} \\ 0 & \frac{16}{153} & \frac{14}{17} & \frac{11}{153} \\ 0 & -\frac{55}{612} & \frac{39}{68} & \frac{79}{153} \end{pmatrix}$$

Por último, en el tercer ejemplo, si $\beta 1 = 1$, $\beta 2 = -10$ y $t = 1/2$, debemos calcular Matrices[1,-10,1/2] y obtenemos:

$$\begin{pmatrix} \frac{11}{12} & -\frac{3}{4} & \frac{5}{6} & 0 \\ \frac{1}{3} & -\frac{1}{2} & \frac{7}{6} & 0 \\ \frac{5}{12} & -\frac{3}{4} & \frac{4}{3} & 0 \\ \frac{5}{24} & \frac{1}{8} & \frac{13}{24} & \frac{1}{8} \end{pmatrix} \quad \begin{pmatrix} \frac{1}{8} & \frac{13}{24} & \frac{1}{8} & \frac{5}{24} \\ 0 & \frac{4}{3} & -\frac{3}{4} & \frac{5}{12} \\ 0 & \frac{7}{6} & -\frac{1}{2} & \frac{1}{3} \\ 0 & \frac{5}{6} & -\frac{3}{4} & \frac{11}{12} \end{pmatrix}$$

No Foto Realismo con Esgrafiado

R. Guerrero *
Lab. de Investig. y Desarrollo en Intelig. Computacional (LIDIC)
Dpto. de Informática - Univ. Nacional de San Luis - San Luis, Argentina
e-mail: *rag@unsl.edu.ar*

Francisco Serón Arbeloa
Grupo de Computación Gráfica Avanzada (GIGA)
Dpto. de Cs. de la Computación - Univ. de Zaragoza - Zaragoza, España
e-mail: *seron@unizar.es*

Abstract

A visual language is used when creating an image to communicate a subject. We can visualize our thinking as we can verbalize it. A diagram, a map, a painting are different examples of our visual language use. Line, shape, color, movement, texture, among others are their structural units, and pictorial art have shown their power for communicating messages without words.

The Non Foto realism area studies the use of expresion and style in the creation of images for communicating visual messages. At this work, we present a prototype tool for the simulation of the artistic technique called *Esgrafiado*. The technique gathers elements from computer graphics, image processing, art and and experimental cognitive psychology trying to give a means for the development of visual communication skills to a user-observer. The technique enables the given concepts could be applied by the user into non artistic fields with specific subjects tools.

Keywords: Computer Graphics, Non Foto Realism, Image Processing, Cognitive Psicology, Art.

Resumen

Cuando se crea una imagen para comunicar una idea, se está utilizando un lenguaje visual. Podemos "visualizar" nuestro pensamiento de la misma manera en que lo verbalizamos. Un diagrama, un mapa, una pintura son ejemplos del uso del lenguaje visual. Sus unidades estructurales son la línea, la forma, el color, el movimiento, la textura, entre otros. El arte pictórico ha demostrado que el uso de dichas unidades permiten transmitir significados sin utilizar palabras.

El área de No Foto realismo estudia la generación de imágenes que permitan la transmisión de mensajes visuales con expresión y estilo. En este trabajo, se presenta un prototipo de herramienta para la simulación de la técnica artística del Esgrafiado. La misma reúne elementos provenientes del área de la computación gráfica, el procesamiento de imágenes, el arte y la psicología cognitiva experimental, intentando brindar un medio para el desarrollo de habilidades de comunicación visual en un usuario-observador así como también lo habilita al uso de los conceptos con otras herramientas para propósitos específicos a ámbitos no artísticos.

Palabras claves: Computación Gráfica, No Foto realismo, Procesamiento de Imágenes, Psicología Cognitiva, Arte.

*Grupo soportado por la UNSL y ANPCYT (Agencia Nac. para la Prom. de la Ciencia y Tec.)

1. INTRODUCCIÓN

El rendering No Foto Realista (NFR) ha concentrado la atención de la comunidad de Computación Gráfica debido al número cada vez más creciente de áreas de aplicación; llendo desde el arte, pasando por el entretenimiento hasta las ilustraciones y su animación (*cartoons*). En los últimos años, un conjunto de conceptos y algoritmos importantes han sido desarrollados proveyendo técnicas de plasmado en estilos diferentes a los logrados por medio del foto realismo.

Las imágenes foto realistas son el producto de aplicar métodos y algoritmos basados en la simulación de leyes físicas para lograr la síntesis de una imágen la cual, en el mejor de los casos, puede ser confundida con la foto de objetos y escenarios reales. Para ello, la simulación de la interacción de la luz con los objetos de una escena, asi como también la textura de los mismos, juegan un papel predominante [5]. Por otro lado, el NFR se concentra en el plasmado interpretativo, artístico, con estilo y expresión; las imágenes resultantes a menudo contienen elementos arbitrarios, ambiguos, aleatorios, en lugar de apegarse a simulaciones precisas de la geometría y la iluminación. No obstante su notoriedad, no existe una única definición de lo que NFR significa pues resulta difícil describirlo como el negativo de otro. Una breve discusión sobre la terminología y las raíces históricas del foto realismo y el no foto realismo es realizada por [9].

La mayoría de los sistemas NFR generan imágenes a partir de la simulación de trazos o pinceladas. Las pinceladas son el procedimiento más común dentro de los estilos pictóricos y se caracterizan por permitir depositar un medio cualquiera (acuarela, óleo, acrílico, etc.) a través de un elemento (pincel, bruñidor, dedo, etc.) sobre un sustrato (madera, vidrio, tela, etc.), dejando bosquejada la ruta que siguió el elemento sobre el sustrato. Una revisión de las técnicas desarrolladas es provista por [12] y más recientemente por [8] y [18]. En los últimos años, los trabajos se han centrado en el perfeccionamiento de técnicas ya desarrolladas [2, 6, 1, 3] o en el desarrollo de nuevas técnicas que permitan la comunicación de un concepto en una manera más comprensible mediante el uso combinado de principios perceptuales, técnicas artísticas y principios de diseño y composición [11, 7, 17]. Otros desarrollos se han focalizado en simplificar la complejidad de los sistemas, intentando disminuir sus tiempos así como también su implementación en hardware [19, 16, 14, 15, 13].

En la sección 2 del presente trabajo se presentan los conceptos básicos de la técnica de Esgrafiado Artístico basada en el principio inverso al de las ya tradicionalmente implementadas. La sección 3 se introduce un prototipo de herramienta para la simulación de dicha técnica. La sección 4 se muestran algunos plasmados no foto realistas logrados a través de la técnica y finalmente, en la sección 5 se bosquejan algunas conclusiones sobre los logros alcanzados.

2. EL ESGRAFIADO ARTÍSTICO

El Esgrafiado es una antigua técnica artística utilizada por las civilizaciones del siglo IV AC. La palabra esgrafiado viene del italiano ***sgraffiare***: hacer incisiones o rascar con una herramienta especial llamada *grafio*. Es una técnica de dibujo que consiste en hacer incisiones o cortes en el estrato superficial de una pintura o enlucido, de manera que quede al descubierto la capa inferior. Se aplica también sobre objetos de cerámica y aún, en la Edad Media, sobre manuscritos en ilustraciones en oro. Estas se diferencian por el conjunto de elementos involucrados en la composición de la obra, mientras que conservan los mismos procesos para la representación del detalle y focalización de la atención. El enfatizado de las características de la escena dependerá de las características de los elementos con los que dispone el artista.

2.1. Componentes Técnicos: Los Elementos y el Proceso

La técnica se clasifica dentro de las escultopinturas en bajorrelieve; basada en colores planos y a diferentes niveles. Primeramente, capas de material de distinto color se superponen unas a otras. Luego, se raspa parte de algunas de ellas para conseguir una decoración polícroma con un diseño específico. Frecuentemente se utiliza una plantilla que se repite a lo largo y ancho de la superficie a esgrafiar.

Usualmente el artista tiene ya realizado un boceto organizado en cuanto a los colores que aplicará para cada figura y el fondo, y utiliza herramientas cortantes o punzantes (espátulas, cuchillos, formones, gubias, etc.).

La técnica permite una gran cantidad de combinaciones. Se pueden trabajar las figuras por planos de color, por valor de línea realizando grafismos al raspar, creando líneas paralelas que van de un color a otro dando lugar a texturas, también se pueden aprovechar las luces y las sombras que se crean entre las capas, y muchas otras habilitando a un universo de descubrimientos y aplicaciones.

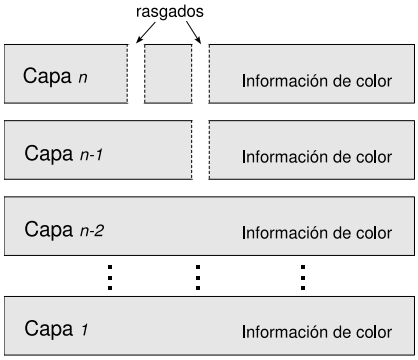


Figura 1: Principio de la técnica.

Si bien la técnica se clasifica como escultopintura en bajorrelieve, el proceso se adapta perfectamente a los preceptos del arte visual, específicamente a la expresión pictográfica. Se podría entonces categorizar la técnica en función del elemento base sobre el cual se desarrolla: cerámica, muros (enlucidos) y pintura (Fig. 2). En todos los casos, los principios compositivos se conservan, cambiando únicamente los elementos utilizados para la composición. Una forma simplificada de esta técnica es utilizada en los primeros pasos de la enseñanza de la expresión pictórica [4], usualmente denominada *Esgrafiado con Tinta China*; donde se utilizan crayones de color, talco o fécula y tinta china como medio; pincel y punzón o clavo o aguja gruesa como objeto; una hoja de papel blanca como fundamento.

(a) Esgrafiado Romano -cerámica- (b) Esgrafiado Segoviano -muro-. (c) Esgrafiado con Tinta China.
.

Figura 2: Diferentes estilos de Esgrafiado.

3. EL ESGRAFIADO DIGITAL

La herramienta propuesta pretende conservar los preceptos de los tres tipos de esgrafiados artísticos planteados. Se ejecuta a partir de una pila de imagenes digitales, con utensilios de rasgados que rasgan las capas según un patrón de diseño establecido y revelan el color de la capa que se encuentra a una cierta profundidad (Fig. 3). La herramienta permite el uso combinado de diferentes utensilios de rasgado.

En la técnica tradicional las capas solamente aportan información de color y la estructura del diseño es determinada por el artista. En la técnica implementada el usuario no controla el rasgado desde la interfase; no obstante, la herramienta intenta habilitar a la generación de plasmados que conserven los aspectos de diseño artístico tradicionales. La información correspondiente a la estructura del diseño proviene de la capa en la base de la pila y las capas superiores aportarán la información de color a asociar a las diferentes primitivas.

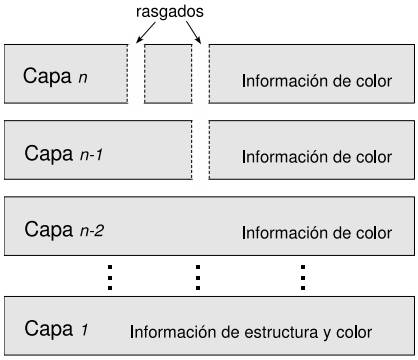


Figura 3: Principio de la técnica.

La información del diseño establece un patrón de rasgado y se define mediante la determinación de un conjunto de atributos importantes a recuperar desde la imagen en la base de la pila. La técnica asume que el usuario ha definido los atributos importantes de la imagen como una función *F(x,y)*, denominada *Función de Importancia* de la manera planteada por [10]. El usuario puede combinar diferentes funciones de importancia asociadas a diferentes características de diseño. Cada función de importancia se asocia con un tipo de rasgado en particular.

El usuario puede también, especificar las características de plasmado de los rasgados. Así, los rasgados representan *primitivas de plasmado* con cualidades de rasgado asociadas -apariencia y profundidad-, traduciéndose en características colorimétricas reveladas luego del rasgado -ver fig. 3-. La imagen resultante mostrará la composición de color producto de combinar: el color de la capa superior, color de las capas reveladas por los rasgados según la disposición establecida por la función de importancia y las características propias de cada primitiva.

3.1. Componentes Técnicos: Los Elementos y el Proceso

Según lo expresado, la técnica se modela bajo los lineamientos de permitir al usuario:

- Especificar cualquier atributo importante o combinación de atributos de una imagen.
- Tener control sobre el número de rasgados generados por el objeto.
- Controlar las características locales de la primitiva (orientación, tamaño, color).
- El uso combinado de diferentes objetos de rasgados (primitivas de plasmado).

3.1.1. Determinación de los atributos

El proceso intenta realizar una detección automática de aquellos atributos importantes procurando preservar los principios de composición artística tradicional -formas, luces y sombras, perspectiva y orden-. Los atributos se especifican mediante funciones de importancia que intentan recuperar la información estructural de diseño desde la imagen en la base de la pila. La función se establece segun lo pautado en [10], permitiendo valorar una región de la imagen acorde con la presencia del atributo deseado. La función retorna un valor alto si la región es muy importante y un valor bajo si la región es de menor importancia. Los valores que surgen

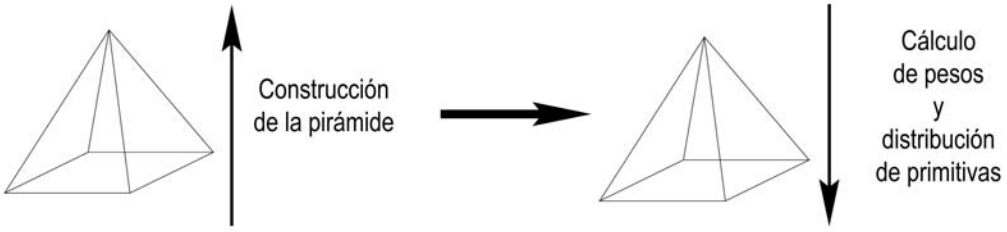


Figura 4: Proceso total. Primeramente se construye la pirámide de multiresoluciones (base al tope), luego se calculan los pesos y se distribuyen las primitivas (tope a la base).

de evaluar la función sobre la imagen original constituyen el nivel más bajo de una pirámide de multiresolución de atributos (ver siguiente sección -ver Fig. 4-). Una vez completada la pirámide, ésta está lista para ser usada en la distribución de los rasgados.

3.1.2. Disposición de los rasgados

Inicialmente, el sistema visual humano realiza una interpretación global de la imagen antes de analizarla en forma localizada. Por ello, es necesario asegurar no corromper la representación global y local de la imagen -conservar principalmente las formas-.

El usuario sólo especifica los atributos a conservar, el resto de las decisiones deben ser elaboradas automáticamente por el proceso. Es decir, la herramienta debe ser capaz de analizar la imagen a una escala global y local con el fin de determinar cómo se realizará el plasmado de los rasgados. Con este fin, la técnica propuesta utiliza una representación a múltiples resoluciones de la imagen para determinar la ubicación global y local de las primitivas durante el proceso de rasgado.

Los atributos importantes a ser preservados, y que luego determinarán la disposición de los rasgados, se procesan generando una pirámide de mulitresolución de atributos. La pirámide permite estimar la interpretación global de la imagen estableciendo y preservando la importancia relativa de los pixeles asociados a los atributos. Así, áreas de la imagen que poseen la misma importancia serán plasmadas con el mismo nivel de detalle en el plasmado.

Por otro lado, los niveles de interpretación local se pueden lograr ajustando localmente las características de las primitivas con el objetivo de crear diferentes efectos visuales al mismo tiempo que se preserva la importancia relativa en toda la imagen.

Cada atributo particular preservado genera su pirámide asociada y consecuentemente se relaciona con un tipo específico de primitiva de rasgado. Es decir, diferentes atributos serán plasmados a diferentes niveles de detalle y con diferentes estilos de rasgado.

3.1.3. Características de los rasgados

Una vez establecida la disposición de los rasgados, la herramienta permite el uso combinado de diferentes utensilios de rasgado -puntos, agrupaciones de puntos, segmentos de línea y agrupaciones de segmentos-. Al momento del plasmado, dichos utensilios se asocian a características de color, tamaño y orientación. Dichas características pueden ser alteradas y acondicionadas según diferentes criterios provenientes de la imagen original, las capas asociadas a los rasgados o criterios propios del usuario.

4. LOS RESULTADOS

Los resultados obtenidos surgen de la manipulación de los parámetros de la herramienta. Cuatro parámetros controlan la técnica: la función de importancia, la cantidad de primitivas a utilizar, el tipo de primitivas y sus ajustes locales; los cuales pueden ser alterados en diferentes combinaciones permitiendo obtener resultados no foto realistas.

Uno de los objetivos principales de la técnica es lograr nuevas representaciones pictóricas conservando o resaltando, según sea conveniente, las características estructurales del diseño. En consecuencia, diversas interpretaciones pueden ser inducidas por el usuario mediante la manipulación de efectos globales y locales durante la construcción de las imágenes a través de la variación de los parámetros mencionados. A continuación se muestran las imágenes logradas con la alteración de cada uno de los parámetros en forma individual. La última sección muestra los resultados obtenidos con la alteración conjunta de los parámetros.

4.1. Imagenes originales

Las imágenes de la figura 5 serán utilizadas como imágenes fundamento para ilustrar los conceptos de colorimetría y estructura desarrollados. La primer imagen es una paleta de colores básica. Las imágenes siguientes corresponden a una imagen sintética generada por computadora -busto-, y la imagen de Lena -típica foto utilizada en Comp. Gráfica-. Ellas son representativas de diferentes composiciones estructurales y colorimétricas producto de su origen.

(a) Paleta básica de colores. (b) Rendering de modelo 3D. (c) Fotografía típica de Lena.

Figura 5: Imágenes utilizadas como sustrato para la presentación de los conceptos.

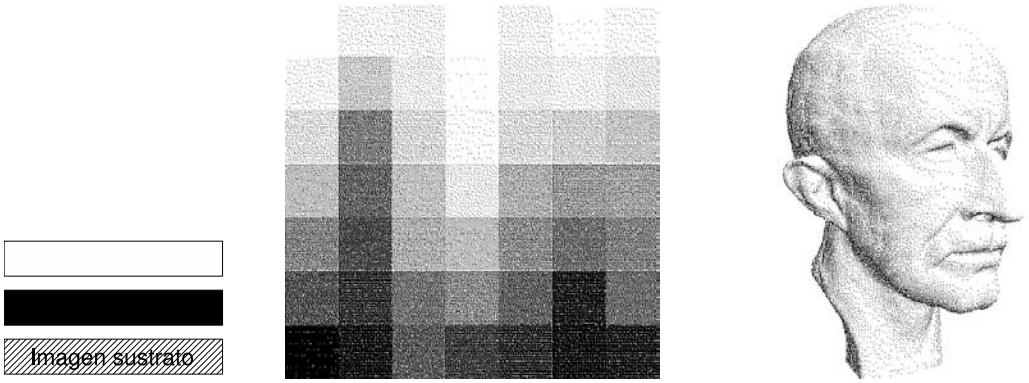
Con la intensión de clarificar un poco más los conceptos abordados, se ha incorporado junto a las figuras presentadas, una representación de la combinación de capas-sustrato involucradas.

4.2. Función de Importancia

El primer parámetro a ser controlado por el usuario es la Función de Importancia. La función de importancia permitirá al usuario especificar los atributos importantes que deben ser preservados. La función establece la distribución global y local de las primitivas a lo largo de todo el proceso, plasmando un mayor número de primitivas en regiones de mayor importancia.

4.2.1. Luces y Sombras

La luminancia de las imágenes permite identificar zonas de luces o sombras que luego pueden utilizarse en forma aislada o combinada. Si el atributo importante es la intensidad f en la posición (x, y), la función de importancia F en la posición (x, y) será: $F(x, y) = 255 - f(x, y)$ previa conversión de los valores de color correspondientes a un único valor de intensidad. La función definida de esta manera considera que las sombras deben plasmarse con un número mayor de atributos, por lo tanto otorga al color negro el mayor peso en la escala.



(a) Capas sustrato. (b) Paleta básica de colores. (c) Modelo 3D.

Figura 6: Luminancia como Función de Importancia.

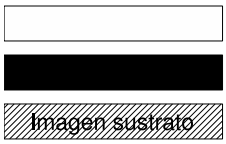
La figura 6 muestra lo expresado. Las imágenes 5(a), 5(b) aportaron la información estructural de luminancia. Una imagen totalmente blanca y otra totalmente negra han aportado la colorimetría para los rasgados. La imagen superior de la pila es la de color blanco. Los rasgados son simples puntos cuya profundidad llega hasta la capa intermedia -imagen negra- y son ubicados según la información establecida por la función de importancia donde, las regiones de sombras son plasmadas con un número mayor de primitivas.

En las imágenes puede apreciarse que la función de importancia definida preserva las características de reflexión de la luz de los colores, agrupándolos acorde con el poder de luminancia que poseen; mayor luminancia para los colores claros, menor para los colores oscuros. En consecuencia la función de importancia permite identificar las zonas de luces y sombras definidas en la imagen sustrato a través de los colores.

4.2.2. Bordes y Formas

Las formas pueden determinarse mediante la detección de los bordes. Los bordes se determinan encontrando las regiones de alto contraste de una imagen. Las regiones de alto o bajo contraste pueden ser localizadas por medio de la detección de discontinuidades en las intensidades. La detección de discontinuidades se realiza a través de las derivadas espaciales de los valores de una imagen. Esta tarea puede ser realizada de diversas maneras: ya sea utilizando operadores de la primer derivada -varianza muestral, gradiente, entre otros-; o utilizando operadores de la segunda derivada -Laplaciana de la Gaussiana-. Cada uno de ellos brindará mayor o menor detalle de información dependiendo del tipo de imagen analizada; esta cualidad puede luego ser utilizada acorde con fines específicos.

Las imágenes de la figura 7 surgen de la imagen del modelo 3D (fig. 5(b)) luego de ser procesada utilizando como función de importancia el gradiente de Sobel (fig. 7(b)) y la varianza



(a) Capas sustrato.

(b) Func. Imp.: Sobel.

(c) Func. Imp.: Varianza.

Figura 7: Bordes como Función de Importancia.

(fig. 7(c)) respectivamente. En ellas puede observarse que la función gradiente detecta cambios en la orientación del vector intensidad que aportan mayor información estructural de formas. Dicha información no solamente se limita a la detección de bordes, sino también de información específica a la imagen analizada -para el caso, rasgos de la cara-, que puede utilizarse convenientemente según la finalidad del plasmado bosquejado.

4.2.3. Combinación de Funciones de Importancia

El usuario puede establecer diferentes atributos de importancia asociados a diferentes funciones las cuales pueden combinarse en una única función. En la figura 8 se han combinado las funciones de varianza e intensidad mostradas en las figuras 6(c) y 7(c) conformando la función F en la posición (x, y): $F(x, y) = 0{,}7 \times FIntens(x, y) + 0{,}3 \times FVarianza(x, y)$

Figura 8: Combinación de Funciones.

En la figura 8 las primitivas se distribuyen en un 70 % para el plasmado de la intensidad y un 30 % para el plasmado de los bordes. Las capas de imágenes corresponden a la mostrada en la figura 7(a). Las características se rasgan con el mismo tipo de primitiva -punto- y las primitivas tienen la misma propiedad de profundidad de rasgado -hasta la capa de color negro-. El número de primitivas totales ha sido calculado en forma proporcional, con el objetivo de conservar las características obtenidas con la aplicación de las funciones en forma individual.

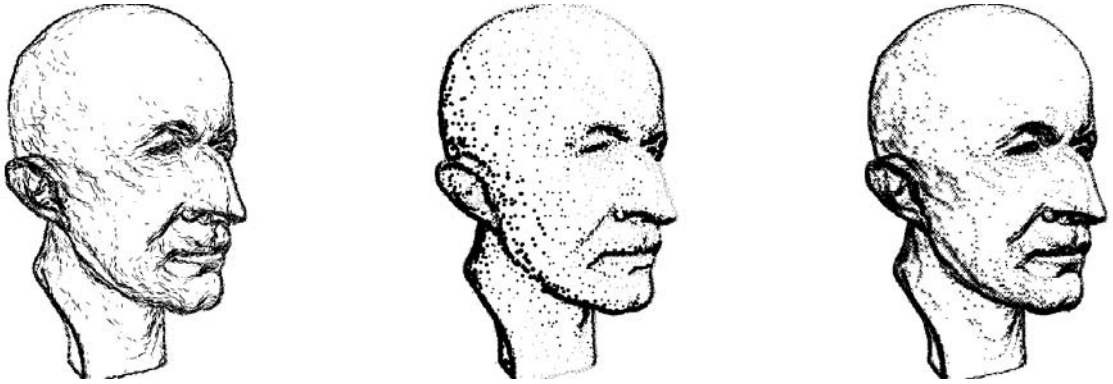
La imagen resultante posee bordes perfectamente delineados -información aportada por la función varianza- al mismo tiempo que se resaltan los detalles de luces y sombras del objeto modelado sintéticamente -información aportada por la función de intensidad-. De esta manera, cualquier número arbitrario de funciones de importancia pueden ser combinadas para formar una única nueva función de importancia. El peso asignado a las funciones individuales refleja el nivel de importancia otorgado a los diferentes atributos de la imagen.

4.2.4. Tipo de Primitivas y Ajustes locales

El tercer parámetro de la técnica es el tipo de primitiva a utilizar. Las herramientas de rasgado, luego de ser aplicadas revelarán el color de la capa subyacente al rasgado. El tipo de rasgado

define la manera en que el medio -pintura- será plasmado, convirtiéndose en una primitiva de plasmado. Las primitivas se ubicarán en las posiciones de la imagen determinadas por la función de importancia. Las primitivas implementadas son: un único punto o pixel, un segmento de línea, una agrupación de puntos conservando un ordenamiento circular, o una agrupación de segmentos.

El cuarto parámetro es el de ajustes a las primitivas. Dependiendo del tipo de primitiva utilizado, se pueden realizar cambios locales en sus características propias (tamaño -escalado, umbralado-, orientación -rotación-, y color -profundidad-) permitiendo introducir efectos de textura local en la imagen, los cuales afectarán su apariencia global. Estos cambios pueden realizarse mediante criterios preestablecidos por el usuario desde la interface o interpretados automáticamente acorde con información de la imagen sustrato analizada.



(a) Func. Imp: Varianza. Primitiva: segmentos de línea de tamaño fijo y rotados.

(b) Func. Imp: Laplaciana de la Gaussiana. Primitiva: grupos de puntos escalados.

(c) Func. Imp.: Varianza. Primitiva: grupo de segmentos conformando una equis.

Figura 9: Diferentes tipos de primitivas y sus ajustes locales.

Las imágenes de la figura 9 muestran el resultado de utilizar diferentes tipos de primitiva con alteraciones de orientación y tamaño. En la imagen 9(a) los segmentos tienen una longitud fija especificada por el usuario pero orientados según el gradiente local al pixel en la imagen original. La imagen muestra un aspecto de bosquejo artístico con un mayor acabado en los rasgos de expresión de la cara del modelo. En la imagen 9(b) la función de importancia aplicada es la Laplaciana de la Gaussiana y las primitivas son agrupaciones de puntos escalados acorde con la intensidad promedio de la región donde debe plasmarse la primitiva

El uso de agrupaciones de segmentos de línea permite generar aspectos de textura en las imágenes, los cuales pueden ser controlados mediante el número y tipo de primitiva seleccionado. En la imagen 9(c), el conjunto de segmentos simulan una "equis" las cuales al momento del plasmado se agrupan entre sí logrando no solamente una impresión de textura sino también conservando las características de luces y sombras de la imagen original -obsérvese la oreja y pómulo de la cara del modelo digital 3D- Los grupos estan escalados en tamaño acorde con la intensidad promedio de la región donde debe plasmarse la primtiva.

4.3. Alterando múltiples parámetros

Las secciones anteriores han mostrado algunos efectos logrados mediante la manipulación individual de los parámetros. En esta sección se muestra el resultado de la manipulación conjunta de los mismos. Para la composición de la imagen resultante -Fig. 11- se ha construido

una pila de cinco imágenes -Fig. 10-. La imagen en el tope de la pila corresponde a la foto original de Lena en escala de grises e invertida -Fig. 5(c)- . La segunda imagen es una imagen totalmente negra, luego le sigue la imagen que aportará información de color para el plasmado del sombrero, las plumas y la boca -Fig. 11(b)-. Posteriormente se encuentra una imagen totalmente naranja y por último, en la base de la pila, se encuentra la imagen original -Fig. 5(c)-, la cual aportará, en este ejemplo, solamente la información estructural.

Figura 10: Capas.

El proceso combina tres funciones de importancia -intensidad, gradiente y Laplaciana de la Gaussiana- y un único tipo de primitiva -punto-. Cada función se asocia con una capa de la pila de imágenes. La capa negra establece el color para la función de importancia de la Laplaciana de la Gaussiana -plasmado de bordes-, la capa de la imagen patrón aporta el color a la función de importancia gradiente -coloreado de plumas, sombrero y boca-, y la capa naranja establece el color para la función de importancia de la intensidad -cabello, sombra de ojos, marco del espejo-. Los colores se plasman en el orden en que se han especificado las capas, desde la base de la pila hacia el tope. La imagen resultante muestra la foto de Lena con una impronta de sepia, donde los elementos de segundo plano se encuentran en color gris; los elementos que acompañan a la composición -cabello y marco del cuadro- en un suave naranja-; los rasgos de la cara se encuentran resaltados en negro y los tres elementos básicos de la composición que encuadran la cara -sombrero, plumas y boca- resaltados en color.

(a) Imagen frontal. Foto de Lena en grises invertidos.

(b) Imagen intermedia. Patrón de colores para el plasmado del gradiente.

(c) Imagen resultante. Combinacion de capas y funciones de importancia.

Figura 11: Fotografía de Lena combinando varias capas colorimétricas y funciones de importancia.

5. CONCLUSIONES

Este trabajo presentó un prototipo de herramienta para la simulación de la técnica artística del Esgrafiado. La misma reúne elementos provenientes del área de la computacion gráfica, el procesamiento de imágenes, el arte y la psicología cognitiva experimental.

La herramienta permite generar plasmados pictóricos desde un punto de vista opuesto al de las tradicionales pinceladas. La técnica desarrollada deja a criterio del usuario cómo plasmar mejor un atributo específico de una imagen, y no intenta identificar automáticamente qué tipo de primitiva debería ser usada para el plasmado de diferentes atributos de la imagen. La apariencia estética y el fin último de una imagen es determinado por el usuario.

Si bien acorde con los aspectos perceptuales, gestálticos y artísticos, algunos componentes básicos de una imagen son más importantes que otros, no se han desarrollado aún reglas estándares para identificar cuáles son los atributos más importantes acorde con nuestra propia interpretación. Es claro que algunos atributos son importantes y deben ser conservados (por ej. bordes), sin embargo, una vez que todos los componentes de una imagen son combinados en el plasmado, resulta difícil establecer cuáles atributos deben ser conservados y cuáles rechazados. La herramienta desarrollada demuestra cuan mediada puede ser nuestra experiencia visual proveyendo una plataforma para la generación de composiciones con características artísticas y diferentes cualidades informativas a partir de una misma información estructural.

Como principio básico del área de No Foto realismo, todas las investigaciones realizadas pretenden generar desarrollos que mejoren la transmisión de un concepto mediante la generación de nuevas técnicas o la simulación de técnicas ya existentes. No obstante, ninguno de estos desarrollos, a conocimiento del autor, intenta ser una herramienta para el aprendizaje de nuevas formas de expresión visual, sino que son, la mayoría de ellas, herramientas con un finalidad específica, sin dejar mayor margen de evolución en el aprendizaje de la expresión visual. La herramienta presentada brinda un medio para el desarrollo de habilidades de comunicación visual en un usuario-observador y lo habilita al uso de los conceptos con otras herramientas para propósitos específicos a ámbitos no artísticos.

No obstante, durante el desarrollo de la misma han surgido numerosas variantes alternativas de extensión a la técnica implementada, tales como: la generación de una interfase de usuario que modele la metáfora de un estudio de dibujo, la ampliación del conjunto de herramientas de esgrafiado implementadas, la simulación de algunos estilos no foto realistas bajo los preceptos de la técnica implementada, habilitar a la configuración de una composición incorporando conceptos tales como "centro de interés" y "focalización", la deducción automática del tipo de primitiva más adecuado para plasmar una característica específica de la composición.

REFERENCIAS

[1] Pascal Barla, Simon Breslav, Joëlle Thollot, and Lee Markosian. Interactive hatching and stippling by example. Technical report, INRIA, 2006.

[2] Pascal Barla, Joëlle Thollot, and François Sillion. Geometric clustering for line drawing simplification. In *Siggraph technical sketch: SIGGRAPH'2005*. ACM, 2005.

[3] Adrien Bousseau, Matthew Kaplan, Joëlle Thollot, and François Sillion. Interactive watercolor rendering with temporal coherence and abstraction. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*. ACM, 2006.

[4] José María Fernández. *Dibujo. Técnicas Artísticas. Fundamentos, Materiales, Técnicas, Ejercicios.* Dastin Ediciones, 2004. ISBN: 8496410560.

[5] J. D. Foley, A. Van Dam, , S. K. Feiner, and J. F. Hughes. *Computer Graphics Principles and Practice*. Addison-Wesley, 1995. ISBN 0201848406.

[6] T. Goetzelmann, K. Ali, K. Hartmann, and Th. Strothotte. Adaptive labeling for illustrations. In *Proc. of 13th Pacific Conference on Computer Graphics and Applications*, volume 196, pages 64–66, 2005.

[7] Amy Ashurst Gooch and Peter Willemsen. Evaluating space perception in npr immersive environments. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 105–110, New York, NY, USA, 2002. ACM Press.

[8] B. Gooch and A. Gooch. *Non-Photorealistic Rendering*. A. K. Peters Ltd., Natick, Massachusetts, 2001. ISBN 1-56881-133-0.

[9] S. Green. Introduction to non-photorealistic rendering. Non-Photorealistic Rendering. SIGGRAPH 99 Course, Chapter 2, 1999.

[10] Roberto Guerrero and John Buchanan. Color importance driven halftoning. In *8vo. Congreso Argentino de Ciencias de la Computación*, volume 2, pages 825–834, Univ. Nac. de Buenos Aires, Buenos Aires, Octouber 2002.

[11] Nick Halper, Stefan Schlechtweg, and Thomas Strothotte. Creating non-photorealistic images the designer's way. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 97–ff, New York, NY, USA, 2002. ACM Press.

[12] John Lansdown and Simon Schofield. Expressive rendering: A review of nonphotorealistic techniques. *IEEE Computer Graphics and Applications*, 15(3):29–37, may 1995.

[13] Hyunjun Lee, Sungtae Kwon, and Seungyong Lee. Real-time pencil rendering. In *NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 37–45, New York, NY, USA, 2006. ACM Press.

[14] Thomas Luft and Oliver Deussen. Real-time watercolor illustrations of plants using a blurred depth test. In *NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 11–20, New York, NY, USA, 2006. ACM Press.

[15] Morgan McGuire and Andi Fein. Real-time rendering of cartoon smoke and clouds. In *NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 21–26, New York, NY, USA, 2006. ACM Press.

[16] Oscar Meruvia Pastor, Bert Freudenberg, , and Thomas Strothotte. Real-time, animated stippling. In *IEEE Computer Graphics and Applications*, volume 23, pages 62–68, July/August 2003.

[17] Maria Shugrina, Margrit Betke, and John Collomosse. Empathic painting: interactive stylization through observed emotional state. In *NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 87–96, New York, NY, USA, 2006. ACM Press.

[18] Thomas Strothotte and Stefan Schlechtweg. *Non-Photorealistic Computer Graphics. Modelling, Rendering, and Animation*. Morgan-Kaufmann Publishers, San Francisco, April 2002. ISBN 1-55860-787-0.

[19] Matthew Webb, Emil Praun, Adam Finkelstein, and Hugues Hoppe. Fine tone control in hardware hatching. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 53–ff, New York, NY, USA, 2002. ACM Press.

Reconocimiento de Números Manuscritos

José Luis Garbi, Paula Mercado
Facultad de Informática. Universidad Nacional de La Plata
La Plata, Argentina, 1900
{jlgarbi, mercado.paula}@gmail.com

Laura Lanzarini, Claudia Russo
III-LIDI (Instituto de Investigación en Informática LIDI)
Facultad de Informática. Universidad Nacional de La Plata
La Plata, Argentina, 1900
{laural, crusso}@lidi.info.unlp.edu.ar

Abstract

At present, handwritten text recognition still represents a wide source of research.

This paper presents a software tool which belongs to the area of ICR (Intelligent Character Recognition) for the recognition of handwritten integers. A classifier based on feedforward neural networks and a set of techniques belonging to digital image processing area are incorporated to this tool, which make the suitable adaptations over the input image. In this way, a handwritten integer made up by several digits is entered and, as a result, the recognition of each of its elements is obtained.

The results of applying this tool over a UCI repository number base have been successful. It is important to notice that, even though the results presented in this paper exclusively refer to handwritten number recognition, this tool can be applied to the complete set of characters.

Finally, some conclusions are presented together with some future lines of work.

Keywords: Handwritten Character Recognition, Preprocessing, Image Segmentation, Neural Networks

Resumen

En la actualidad, el reconocimiento de texto manuscrito sigue siendo una fuente de intensa investigación.

Este paper presenta una herramienta de software perteneciente al área de Reconocimiento Inteligente de Caracteres (ICR – Intelligent Character Recognition) para el reconocimiento de números enteros manuscritos. En ella se integra un clasificador basado en redes neuronales feedforward y un conjunto de técnicas pertenecientes al área de procesamiento de imágenes digitales que realiza las adaptaciones adecuadas sobre la imagen de entrada. De esta forma, se ingresa un número entero manuscrito formado por varios dígitos y se obtiene como resultado el reconocimiento de cada uno de los elementos que lo componen.

Los resultados de la aplicación de esta herramienta sobre una base de números del repositorio UCI han sido satisfactorios. Es importante destacar que, si bien los resultados expuestos en este artículo se refieren exclusivamente al reconocimiento de números manuscritos, esta herramienta puede ser aplicada al conjunto de caracteres completo.

Finalmente se incluyen algunas conclusiones así como algunas líneas de trabajo futuras.

Palabras claves: Reconocimiento de caracteres manuscritos, Preprocesamiento, Segmentación de imágenes, Redes Neuronales

1. Introducción

En la actualidad, el reconocimiento preciso en textos escritos a máquina se considera un problema resuelto. Sin embargo, no ocurre lo mismo con el reconocimiento de la impresión manual, es decir, aquella que proviene de la caligrafía humana, la cual sigue siendo una fuente de intensa investigación.

Cuando la información es ingresada on-line, es posible aprovechar la velocidad y la dirección de los segmentos trazados como información de entrada [12]. Los asistentes digitales personales o PDA (personal digital assistant) han obtenido excelentes resultados en el reconocimiento de caracteres escritos a mano alzada ya que su software aprovecha este tipo de información. Además, el usuario se puede entrenar y ayudar al dispositivo usando solamente formas específicas de letras. Estos mismos métodos no se pueden trasladar a los programas que se encargan de interpretar los caracteres de documentos escaneados.

Un problema aun más complejo de resolver es el reconocimiento de textos cursivos, es decir aquellos en el que todas las letras se encuentran conectadas formando una palabra [11][2]. En estos casos, la complejidad del problema puede reducirse recurriendo a información adicional ya sea gramatical o contextual. Por ejemplo, el reconocimiento de un conjunto de palabras previamente clasificadas es más fácil de resolver que tratar de analizar, de manera individual, los caracteres de la escritura; independientemente del tamaño del conjunto.

Este artículo presenta una herramienta de software para el reconocimiento de números enteros manuscritos perteneciente al área de Reconocimiento Inteligente de Caracteres o ICR (Intelligent Carácter Recognition). Dicha herramienta integra un clasificador basado en redes neuronales feedforward y un conjunto de técnicas pertenecientes al área de procesamiento de imágenes digitales que realiza las adaptaciones adecuadas sobre la imagen de entrada.

La redes neuronales feedforward, entrenadas mediante una técnica de gradiente, han demostrado ser excelentes clasificadores con una gran tolerancia al ruido naturalmente presente en la entrada de datos. Esto último es una característica importante a la hora de procesar imágenes digitales. Sin embargo, cuando el patrón a reconocer es directamente una imagen, la respuesta de la red es muy sensible al tamaño y posicionamiento del sector a reconocer. Por tal motivo se incluyó en esta herramienta un sistema de preprocesamiento y segmentación de los distintos elementos de la imagen que, junto con un escalado adecuado, ayudarán al clasificador a brindar una respuesta correcta.

Este artículo está organizado de la siguiente forma: en la sección 2 se da una breve descripción del sistema propuesto en este artículo y en las secciones 3,4 y 5 se detallan cada uno de sus módulos y se incluyen los resultados obtenidos. Finalmente la sección 6 contiene las conclusiones así como algunas líneas de trabajo futuras.

2. Descripción de la herramienta de software propuesta

La herramienta propuesta en este artículo trabaja sobre imágenes binarias de números enteros manuscritos. El primer paso es reconocer, dentro de la imagen, los componentes correspondientes a dígitos. Esto se lleva a cabo tomando los componentes conexos de la misma. Dado que habitualmente la imagen de entrada presenta cierto ruido, ya sea porque se encuentra presente en la señal original o porque se produce durante el paso de adquisición de la misma, se utiliza un módulo de procesamiento digital de imágenes para normalizar el dígito y así prepararlo para que sea tomado como entrada del clasificador basado en redes neuronales. Si el clasificador puede determinar de qué dígito se trata, se da por terminada la tarea para ese posible dígito y se continúa con el siguiente. Si el reconocimiento falla, se asume que se trata de dígitos que se encuentran

superpuestos de algún modo y se intenta separarlos mediante un algoritmo de segmentación. Luego de segmentar un dígito, nuevamente se envía al clasificador, previa normalización. La segmentación continúa hasta reconocer un dígito apropiadamente o hasta que se han intentado todas las posibles segmentaciones; en cuyo caso se determina ese posible dígito como no reconocido. La figura 1 representa el algoritmo descripto.

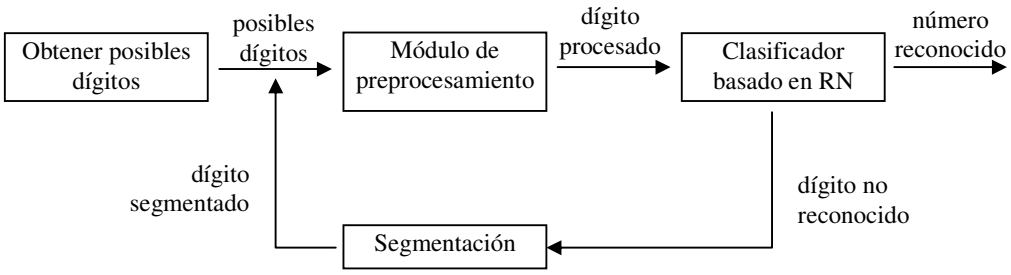


Figura 1: Esquema general de la herramienta de software propuesta

A continuación se describe con más detalle cada uno de los módulos antes mencionados.

3. Módulo de Segmentación

Si bien existen diversos mecanismos de segmentación para dígitos [3][2][11], se seleccionó el método desarrollado por Punnoose denominado EDF (Extended Drop Fall) [9] ya que ha sido utilizado en aplicaciones similares con muy buenos resultados [8].

EDF traza un camino vertical dentro de la imagen desde un punto de partida (x_0,y_0), ubicado en la primera fila, hasta el punto final (x_n,y_n) ubicado en la última fila. Este camino segmenta la imagen en dos porciones; la que se encuentra a su derecha y la que se encuentra a su izquierda.
Existen un conjunto de reglas que determinan el camino trazado. Las reglas son evaluadas en orden. Si una regla falla, se intenta con la siguiente, así hasta la última. Si el camino se encuentra en la posición (x_i, y_i), se evalúan las siguientes reglas en este orden:

REGLA 1:	Si $(x_i + 1, y_i) = 0 \rightarrow$ moverse a $(x_i + 1, y_i)$. Si abajo es negro (el color del fondo), moverse hacia abajo.
REGLA 2:	Si $(x_i + 1, y_i + 1) = 0 \rightarrow$ moverse a $(x_i + 1, y_i + 1)$. Si abajo a la derecha es negro, moverse hacia abajo a la derecha.
REGLA 3:	Si $(x_i + 1, y_i - 1) = 0 \rightarrow$ moverse a $(x_i + 1, y_i - 1)$. Si abajo a la izquierda es negro, moverse hacia abajo a la izquierda.
REGLA 4:	Si $(x_i, y_i + 1) = 0 \rightarrow$ moverse a $(x_i, y_i + 1)$. Si a la derecha es negro, moverse hacia la derecha.
REGLA 5:	Si $(x_i, y_i - 1) = 0$ ^ dirección = derecha$\rightarrow$ moverse a $(x_i, y_i - 1)$. Si a la izquierda es negro y se viene desde la derecha, moverse hacia la izquierda.
REGLA 6:	Moverse a $(x_i + 1, y_i)$. Si fallan todas las anteriores, mover hacia abajo.

En la REGLA 6 es donde se produce el corte, pues se intentaron todos los caminos previos y no hubo posibilidad de seguir por el color de fondo. El algoritmo EDF, intenta moverse por el color de fondo lo más que puede. Una vez que ingresa a píxeles pertenecientes al color del dígito, aplica el mismo conjunto de reglas tratando de mantenerse dentro de este color, hasta que no tiene otra

alternativa que volver al color de fondo o hasta que llegó a la última fila; en donde se termina la segmentación. La figura 2 muestra el resultado de aplicar EDF entre dos dígitos cero unidos, situación que se encuentra comúnmente en los números manuscritos.

Original con trazo de segmentación | Porción izquierda | Porción derecha

Figura 2: Segmentación EDF implementada

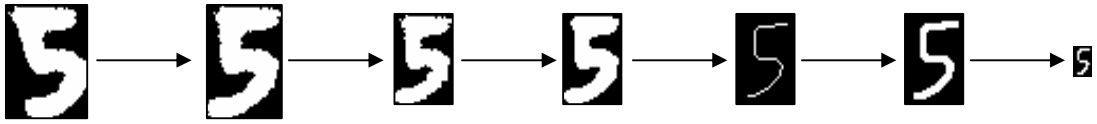
4. Módulo de preprocesamiento

El objetivo del módulo de preprocesamiento es adaptar o normalizar apropiadamente los dígitos del número entero para que sean entradas adecuadas de las redes neuronales del clasificador.

La escritura manuscrita presenta varios desafíos a la hora de ser considerada como entrada en un sistema de reconocimiento. En sistemas de reconocimiento óptico de caracteres (OCR), cada caracter puede variar en tamaño y fuente. En la escritura manuscrita, además de estas complicaciones, se encuentran las relacionadas al tipo de lápiz, lapicera o marcador utilizados, pues cada uno produce un grosor y una terminación diferente que influye en proceso de escaneo. Además se deben considerar los problemas de la inclinación de la letra manuscrita y, como ya se ha dicho antes, la unión de números que ocurre comúnmente durante la escritura; que deben ser segmentados.

Adicionalmente, el reconocedor debe tener en cuenta las diferentes formas de escribir un mismo número. Por ejemplo, existen diferentes formas de escribir los dígitos 1, 4 o 7. Estas diferencias son comunes dentro de una misma población y se acentúan más entre poblaciones con culturas diferentes. De todas formas, esta consideración forma parte del entrenamiento del sistema con un conjunto de dígitos apropiados a la población en consideración.

El preprocesamiento a realizar consiste de los siguientes pasos: a) Corrección de la inclinación, b) Redimensionamiento, c) Filtrado, d) Adelgazamiento, e) Ensanchamiento, f) Reducción del tamaño



Original a) b) c) d) e) f)

Figura 3: Pasos realizados por el módulo de preprocesamiento

A continuación se explicará cada uno de estos pasos.

4.1. Corrección de la inclinación

Comúnmente, cualquier texto escrito a mano presenta cierta inclinación. La inclinación de un dígito influye en su reconocimiento. Para solucionar este problema, debe calcularse la inclinación de la imagen y corregirla en la dirección apropiada.

Para determinar el ángulo de inclinación θ se utiliza la siguiente fórmula, derivada a partir del cálculo de los momentos generales invariantes [7].

$$\theta = \frac{\tan^{-1}\left(\frac{2\,\mu_{11}}{\mu_{20}-\mu_{02}}\right)}{2}$$

donde se define

$\mu_{11} = m_{11} - m_{10}m_{01}/m_{00}; \mu_{20} = m_{20} - m_{10}^2/m_{00}; \mu_{02} = m_{02} - m_{01}^2/m_{00}$

siendo $m_{pq} = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} x^p y^q I(x,y); p,q = 0,1,...\infty;$ el momento discreto p q de la imagen binaria discreta I, de tamaño NxM.

Una vez determinado el ángulo θ es necesario corregir la inclinación para lo cual se consideraron dos alternativas: rotar la imagen según el ángulo calculado o realizar un corrimiento horizontal de los píxels utilizando el método descripto en [5]. La segunda alternativa mantiene intactos los píxeles de la fila inferior y luego cada píxel de las filas superiores se desplaza un determinado número de posiciones según su altura logrando un dígito no inclinado. Para determinar la nueva posición de un píxel se utiliza la siguiente fórmula.

$\forall I(x,y) = 1,\ I'(x',y') = 1, siendo\ x' = x - y\tan(\theta),\ y' = y,$ siendo I la imagen original e I´ la imagen resultado.

En definitiva, lo que se logra es un corrimiento horizontal de filas. A medida que se incrementa la altura de una fila el desplazamiento será mayor. Los resultados de ambos métodos de corrección de la inclinación se pueden observar en la figura 4.

Original | Corrección con rotación | Corrección implementada

Figura 4: Mecanismos de corrección de la inclinación analizados para el desarrollo de la herramienta

Para la implementación de la herramienta propuesta en este artículo se seleccionó la segunda opción, pues genera una imagen de mejor calidad.

4.2. Redimensionamiento

El cambio de tamaño de la imagen original independiza a las redes neuronales del tamaño de escritura y de la resolución del escáner. Además, los algoritmos posteriores de procesamiento digital tienen un mejor rendimiento sobre imágenes más pequeñas.

La estandarización de la relación de aspecto es un paso previo necesario para la modificación de tamaño que sufrirá la imagen. El tamaño final de la imagen preparada para ingresarse a la red neuronal es de 13x9 y su relación de aspecto es 1.44 (13/9 = 1.44). El ancho de la imagen es aproximadamente el 69% de su altura. Se seleccionaron estos valores teniendo en cuenta la relación de aspecto media del conjunto de dígitos de la base de datos con la que trabajamos. Pero antes de llegar a este tamaño, la imagen original se modifica a una de 39x27 píxeles. La relación de aspecto se mantiene y el tamaño es adecuado para el procesamiento restante, aunque es demasiado grande para ser la entrada de la red neuronal.

Para la modificación de tamaño se requiere, en primer lugar, centrar el dígito teniendo en cuenta la relación de aspecto 1.44, en caso de ser necesario. Luego se lleva la imagen al tamaño de 39x27 píxeles. Existen diferentes algoritmos para modificar el tamaño de una imagen. Entre los más conocidos podemos mencionar a la interpolación del vecino más próximo, lineal, bilineal y bicúbica, entre otras [5]. Como se están utilizando imágenes binarias, se seleccionó la interpolación del vecino más próximo ya que sus resultados no generan nuevos valores de píxeles.

4.3. Filtrado

La imagen filtrada se diferencia de la original en que se ha logrado detallar y resaltar sus píxeles en los lugares donde más lo necesita. Muchas veces, al escanear un dígito se producen imperfecciones debido al ruido y características del trazo con el que se hizo el número. Estas imperfecciones merman la capacidad de detección de las redes neuronales, lo que hace imperioso intentar reducirlas. Con este objetivo en mente, se filtró la imagen de 39x27 píxeles con el filtro de la mediana de tamaño 3x3 [5].

4.4. Adelgazamiento

El adelgazamiento es un paso fundamental en las etapas del procesamiento digital. Esta fase independiza a la red neuronal del grosor con el que fue escrito el número. Su objetivo es encontrar una imagen binaria del grosor de un píxel que respete de la mejor manera posible las características morfológicas del dígito original. La misma debe estar contenida en la imagen y conservar la conectividad.

Existen diferentes algoritmos para obtener este resultado. Una manera es obtener el esqueleto de la imagen que está formado por aquellos píxeles que se encuentran equidistantes al menos de dos píxeles del borde. Otra forma de obtener una imagen del grosor de un píxel es utilizando un algoritmo llamado adelgazamiento. Desde el punto de vista de la herramienta propuesta, este algoritmo, que consiste en una eliminación progresiva de los píxeles del borde hasta llegar al grosor deseado, aporta mejores resultados que la obtención del esqueleto. Esto puede observarse en la figura 5.

De todas formas, el algoritmo utilizado para encontrar la representación de grosor mínimo del dígito es el propuesto en [6]. Este algoritmo no siempre produce exactamente un resultado de grosor uno en todos los tramos del dígito, pero es más efectivo en la preservación de la morfología del mismo.

Figura 5: Diferentes resultados de adelgazamiento

4.5. Ensanchamiento

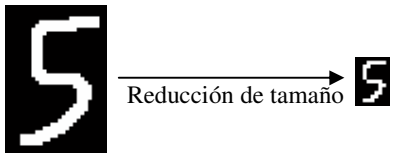
El siguiente paso del procesamiento de la imagen es ensanchar el resultado obtenido en la etapa anterior. Esto logra un grosor uniforme para todos los dígitos que se vayan a reconocer. Además, con este procedimiento se evita perder definición cuando la imagen sea finalmente redimensionada a su tamaño definitivo de 13x9 píxeles; es decir, la tercera parte del tamaño con el que se viene

trabajando. El algoritmo de ensanchamiento es realmente simple. Por cada píxel de la imagen original que pertenece al dígito, sus 8 vecinos son puestos en 1, es decir, que aproximadamente se triplica el grosor de cada línea que conforma al dígito.

4.6. Reducción del tamaño

Como ya se ha dicho anteriormente, el tamaño final de la imagen es de 13x9 píxeles. Este tamaño es adecuado para las redes neuronales y aún preserva mayormente las características del dígito original gracias al procesamiento realizado.

Para llevar la imagen de 39x27 píxeles a una de 13x9, se subdivide la imagen original en regiones de 3x3 píxeles, dando lugar a una matriz de 13x9 regiones. Cada región corresponde a un píxel de la imagen final. El píxel de la imagen reducida será 1 si más de la mitad de los píxeles de la región original correspondiente es 1. En la figura 6 se puede observar el resultado del algoritmo de reducción de tamaño.



39x27 13x9

Figura 6: Reducción de tamaño de la imagen ensanchada

5. Clasificador basado en Redes Neuronales

El clasificador de números manuscritos recibe la imagen preprocesada según lo indicado en la sección anterior y utiliza redes neuronales feedforward para realizar el reconocimiento. Este tipo de redes, adaptadas mediante la técnica del gradiente descendente de manera supervisada, son conocidas como MLP (multilayer perceptron) y proporcionan muy buenos resultados cuando los datos utilizados en el entrenamiento cubren adecuadamente el espacio de entrada. Su convergencia depende de la consistencia de la información que se les suministra.

Por todo lo anterior, y con el objetivo de acotar el tiempo de entrenamiento, se ha utilizado un clasificador basado en un sistema de votación formado por tres redes neuronales feedforward, con idéntica arquitectura, que reconocen en paralelo una misma entrada. Posteriormente, el resultado de cada red es supervisado por una unidad parametrizable que permite especificar el grado de precisión mínimo de la red para que su respuesta sea aceptada. Luego, se utiliza un sistema de votación, que en base al resultado de cada red toma una decisión respecto de qué digito se trata, o si no hay suficiente confiabilidad, determina al dígito como “no reconocido”.

En las secciones siguientes se explicará más en detalle cada parte del módulo de reconocimiento.

5.1. Arquitectura de cada red neuronal

Las tres redes neuronales utilizadas presentan la misma arquitectura; son feedfroward completamente conectadas formadas por: una capa de entrada, una oculta y una de salida.

La entrada de la red neuronal está conformada por 117 entradas independientes correspondientes a cada píxel de la imagen de entrada (de tamaño 13x9).

La capa oculta de cada red está conformada por 50 neuronas. Este número se ha seleccionado en base a estudios empíricos teniendo en cuenta el rendimiento del sistema. Para elegir la cantidad de neuronas de la capa oculta se realizaron distintas pruebas sobre el conjunto de dígitos con 30, 40, 50, 60 y 70 neuronas; para cada una de ellas se utilizaron los siguientes valores para la velocidad de

aprendizaje: 0.005 y 0.0005. Las mediciones realizadas muestran que 50 neuronas en la capa oculta permiten obtener la mejor clasificación en un tiempo de entrenamiento aceptable. El valor elegido para la velocidad de aprendizaje es 0.005.

La capa de salida está conformada por 10 neuronas, una para cada dígito. Cada neurona de salida puede tomar un valor en el intervalo [0,1]. Dada una entrada, se espera que la red genere una salida formada por 10 números reales entre [0,1] donde uno de ellos es notablemente superior al resto y cercano a uno.

5.2. Entrenamiento de las redes neuronales

El entrenamiento de cada red del clasificador se realiza de manera independiente comenzado desde distintos valores aleatorios para sus matrices de pesos.

Los datos utilizados provienen de una combinación de los dígitos de [1] y [4]. La base de datos está conformada por 7620 dígitos en total. A partir de ella, se crearon tres conjuntos, un conjunto de entrenamiento, un conjunto de validación y un conjunto de testeo. El primer conjunto está conformado por el 50% de los dígitos (3809 dígitos). El segundo conjunto con el 15% de los dígitos (1291 dígitos). Finalmente, el conjunto de testeo se conforma con el 35% restante (2520 dígitos).

Para entrenar cada red neuronal utilizada se practica un esquema de "parada temprana". Mediante este enfoque se pretende evitar que la red pierda generalidad. Tradicionalmente una red MLP se entrena tratando de minimizar su error hasta que el mismo sea inferior a cierta cota. Se practica un enfoque de parada temprana para evitar que la red se especialice demasiado en los dígitos contenidos en el conjunto de entrenamiento y así pierda generalidad sobre dígitos que se puedan presentar cuando la red esté en funcionamiento en una aplicación de reconocimiento real.

Teniendo esto en mente, lo que se hace es utilizar el conjunto de validación. El entrenamiento entonces minimiza el error de la red sobre el conjunto de entrenamiento, pero en cada paso verifica que también descienda el error del conjunto de validación. Cuando el error del conjunto de validación en el paso j + 1 sea mayor que en el paso anterior j, quiere decir que la red se está especializando demasiado sobre el conjunto de entrenamiento para reconocer cada patrón y puede estar perdiendo generalidad. En este momento se detiene el entrenamiento. En la herramienta propuesta la medida de error utilizada corresponde al error cuadrático medio del conjunto correspondiente.

5.3. Precisión de la solución

Una vez obtenida la respuesta de una red neuronal en base a un dígito de entrada, las 10 neuronas de la capa de salida son analizadas. Una unidad parametrizable se encarga de garantizar cierto grado de confiabilidad a la respuesta producida por la red neuronal. Cada red MLP tiene su propia unidad de precisión y utiliza dos parámetros. El primero, denominado *UMBRAL*, se aplica al valor más alto entre las respuestas de las neuronas de la capa de salida y determina una cota inferior, para este valor, a fin de que sea considerado válido. De no ser así, el dígito se clasifica como no reconocido.

El segundo parámetro, llamado *DIF*, permite cuantificar la diferencia entre los dos máximos valores de las 10 neuronas de salida. Esto evita que una entrada sea reconocida como dos o más dígitos con un grado similar de confiabilidad. Si esto sucede, nuevamente el dígito se considera no reconocido. En resumen, las reglas utilizadas son las siguientes:

REGLA 1: SI max(salida) < *UMBRAL* ENTONCES no reconocido

REGLA 2: SI max(salida) * *DIF* < segundo_max(salida) ENTONCES no reconocido

5.4. Unidad de votación

La unidad de votación es el módulo encargado de recoger todos los resultados de las redes neuronales y en base a ellos tomar la decisión de definir el dígito en cuestión o clasificarlo como no reconocido.

En la herramienta propuesta se encuentran implementadas dos unidades, llamadas "votacion23" y "votacion100". En la primera de ellas, se toma la decisión de clasificar al dígito de entrada si dos de las tres redes llegan al mismo resultado. "Votacion100" escoge el resultado si las tres redes coinciden respecto a una entrada, o sea, existe un acuerdo del 100%.

Cada módulo de votación determina un comportamiento diferente para el clasificador. Con "votacion23", el sistema es más tolerante, produciendo una mayor cantidad de aciertos en cuanto al número de dígitos reconocidos, pero también un mayor número de falsos positivos, es decir, dígitos reconocidos incorrectamente. Con "votacion100" la cantidad de dígitos reconocidos disminuye, pero también descienden los errores cometidos, incrementándose la cantidad de dígitos no reconocidos. Dependiendo del uso del sistema en casos particulares, los cuales definen la precisión deseada, podría ser preferible tener una mayor cantidad de dígitos no reconocidos, a que estos sean identificados incorrectamente.

5.5. Resultados de la clasificación

Para demostrar el comportamiento del módulo de reconocimiento se presenta la siguiente tabla, en la cual se muestran los resultados de las tres redes neuronales trabajando de manera independiente sobre la base de datos de dígitos, como así también el escrutinio de ambas unidades de votación; siendo UMBRAL = 0.55 y DIF = 0.85.

	Correctas	Rechazadas	Incorrectas
MLP 1	80.60 %	17.78 %	1.63 %
MLP 2	80.99 %	17.62 %	1.39 %
MLP 3	80.36 %	18.10 %	1.55 %
Votacion23	81.71 %	17.14 %	1.15 %
Votacion100	72.74 %	26.71 %	0.56 %

Tabla 1: Comportamiento del módulo de reconocimiento (umbral = 0.55 dif = 0.85)

En la tabla puede observarse, además del comportamiento de cada red individual, las características que antes se explicaron acerca de las unidades de votación. El módulo "votacion23" permite obtener un valor más bajo de incorrectos, teniendo en cuenta el comportamiento de cada red en particular, sin perder efectividad en el porcentaje de dígitos reconocidos correctamente. Con "votacion100" se reduce la cantidad de dígitos correctos, pero lo mismo sucede con los dígitos reconocidos incorrectamente.

6. Conclusiones y líneas de trabajo futuras

Se ha presentado una herramienta de software para el reconocimiento de números enteros manuscritos que integra un clasificador basado en redes neuronales feedforward y un conjunto de técnicas pertenecientes al área de procesamiento de imágenes digitales que realiza las adaptaciones adecuadas sobre la imagen de entrada. Los resultados de la aplicación de esta herramienta sobre una base de números del repositorio UCI han sido satisfactorios.

Es importante destacar, que el uso de clasificadores basados en redes neuronales permite obtener una respuesta en un tiempo computacional breve ya que la etapa de entrenamiento se realiza off-line. Además, el preprocesamiento de la imagen de entrada ha permitido obtener un vector de características adecuado para cada una de las redes neuronales resolviendo los problemas habituales

de posicionamiento y escalado. Finalmente,.el sistema de votación ha demostrado ser efectivo para resolver este tipo de problemas y su respuesta puede ajustarse en función de una cota de error prefijada.

Actualmente se está trabajando en el diseño de un clasificador basado en una red neuronal competitiva que permita incorporar nuevo conocimiento sin necesidad de rehacer el entrenamiento completo. Resulta de interés analizar el impacto que esta modificación pude provocar sobre el tiempo de respuesta.

7. Referencias

[1] Alpaydin E., Kayna C. Optical Recognition of Handwritten Digits. *UCI Machine Learning Repository*. ftp://ftp.ics.uci.edu/pub/machine-learning-databases/optdigits. Último acceso julio de 2007.

[2] Arica, N. Yarman-Vural, F.T. Optical character recognition for cursive handwriting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol 24, nro. 6, pp. 801-813.

[3] R. G. Casey y E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(7), 690{706 (1996).

[4] Duin R. Multiple Features Database. *UCI Machine Learning Repository*. ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mfeat. Último acceso julio de 2007.

[5] Gonzalez-Woods. "Digital Image Processing". Prentice Hall. 2002 2da edición.

[6] A. Gupta, M.V. Nagendraprasad, A. Liu, P. Wang, S. Ayyadurai. "An Integrated Architecture for Recognition of Totally Unconstrained Handwritten Numerals". Massachusetts Institute of Technology 1993.

[7] D. Maravall Gomez-Allende. "Reconocimiento de formas y visión artificial". Addison-Wesley Iberoamericana. 1993.

[8] Palacios R y Gupta A. A system for processing handwritten bank checks automatically. Image and Vision Computing, 2002.

[9] Punnoose J. An Improved Segmentation Module for Identification of Handwritten Numerals. *Department of Electrical Engineering and Computer Science at the MIT.* 1999

[10] Tyan J., Neubauer C. Character segmentation method for vehicle license plate recognition. US Patent 6,473,517, 2002

[11] Yamada H., Nakano Y. Cursive Handwritten Word Recognition Using Multiple Segmentation Determined by Contour Analysis. *IEICE Transactions on Information and Systems*. Vol.E79-D, No.5, pp.464-470. 1996

[12] Yanivello D., Lanzarini L. Reconocimiento de Comandos Gestuales utilizando GesRN. *X Congreso Argentino de Ciencias de la Computación. CACIC 2004*. Universidad Nacional de La Matanza. Bs.As. Argentina. ISBN 987-9495-58-6

Remallado de Terrenos Multirresolución

M. Cifuentes [1,2], A. Corbellini[2], M. Rolón[2], G. Rodríguez[2], P. Lotito[2], A. Clausse[2,3]
cifuente@exa.unicen.edu.ar
[1]Comisión de Investigaciones Científicas de la Provincia de Buenos Aires
[2]Universidad Nacional del Centro de la Provincia de Buenos Aires
[3]Comisión Nacional de Energía Atómica

Abstract

This paper proposes a new methodology for the multiresolution visualization of earth's topography defined by many adjacent real topographical models. The strategy refines progressively the model subdividing it into rectangular triangles, building a binary hierarchy generated by the progressive refining of the model, using the error indicator. It combines the observer location, the local curvature of the zone and the height fields and the triangle size put before. The triangulation applies refining and triangle grouping processes, adding or deleting detail to the representation.

We also apply previously developed ideas to real elevation digital models.

Key words: Multiresolution, topographical model, local curvature, triangle strip, real elevation, DEM (Digital Elevation Model).

Resumen

Se propone una nueva metodología para la visualización multirresolución de topografía terrestre definida por varios modelos topográficos reales colindantes. La estrategia refina progresivamente al modelo subdividiéndolo en triángulos rectángulos construyendo una jerarquía binaria generada por el refinamiento progresivo del modelo, según un indicador de error que combina ubicación del observador, curvatura local de la zona y del campo de alturas y tamaño deseable en los triángulos. La triangulación aplica procesos de refinamiento y agrupación de triángulos, agregando o sacando detalle a la representación. Además, se aplican ideas desarrolladas anteriormente a modelos digitales de elevación reales.

Palabras Claves: Multiresolución, modelo topográfico, curvatura local, triangle strip, elevación real, MED (Modelo de Elevación Digital).

1 INTRODUCCIÓN

En los sistemas de información geográfica, la porción de superficie terrestre a representar proviene de grandes modelos topográficos digitales, razón por la cual es necesario incorporar técnicas que faciliten su ágil manipulación y/o visualización en tiempo real. La literatura actual propone técnicas de generación de modelos multiresolución que posibilitan la extracción del mismo terreno con resolución variada. Por ejemplo, representar con mucho detalle aquellas regiones del terreno que se encuentran cercanas al ojo del observador y modelar con unos pocos polígonos las regiones alejadas [1-8].

Este trabajo propone una estrategia que agiliza la visualización como también la transmisión en tiempo real de extensos terrenos. La idea es construir una aproximación multiresolución de la superficie basada en la ubicación del observador, inclinación de la dirección de vista, e indicador de la curvatura local de la región. Los triángulos que la componen se organizan en un *árbol binario* del que se extrae la triangulación final de la superficie: una malla poligonal multirresolución conforme.

Además, el algoritmo propuesto para la generación de la malla almacena los polígonos en un *triangle strip* como una serie de triángulos conectados que comparten vértices. Esta estructura acelera la carga en memoria, la visualización y disminuye el espacio necesario en disco. Las placas gráficas disponibles la optimizan, constituyendo el modo más eficiente de describir un objeto en *computers graphics*. La compactación comienza después de introducidos los tres primeros vértices del primer triángulo en el strip. Por cada nuevo triángulo se incorpora un nuevo vértice, los dos vértices restantes se comparten con el del último triángulo introducido.

Finalmente, se aplicó para la visualización de modelos poligonales correspondientes a modelos digitales de elevación reales (MDE) y sintéticos, un algoritmo de reconocimiento de vértices colineales el cual minimiza la redundancia de los mismos entre triángulos adyacentes.

2 METODOLOGÍA PARA LA SIMPLIFICACIÓN DE TERRENOS

En la actualidad es una realidad el modelado computacional de topografía terrestre obtenida a partir de modelos digitales de elevación (MDE), grillas regulares compuestas por millones de cotas que describen una porción de la superficie terrestre. Generalmente, para modelar terrenos extensos se agrupan MDEs que definen sectores colindantes de la superficie en estudio. En principio, surge la necesidad de automatizar la fusión de MDEs con la implicancia del tratamiento interactivo de grillas voluminosas con tamaños y/o resoluciones disímiles [9]. En consecuencia, la región geográfica de interés surge como resultado de la aplicación de un *algoritmo de encaje* especialmente implementado y cuya funcionalidad, por ejemplo, facilita la unión de varios MDEs representantes de zonas adyacentes del terreno en estudio. La magnitud de la información de los MDEs constituye un problema al momento de visualizar la topografía; no obstante, es posible reducir este volumen procesando únicamente la cápsula convexa de cotas inmersas en la vista. De todas maneras, es conveniente construir modelos simplificados que empleen un reducido conjunto de cotas que conserven la apariencia en la representación.

Básicamente, la idea es reducir la cantidad de polígonos que definen originalmente al modelo construyendo una aproximación multiresolución. El algoritmo de simplificación refina incrementalmente el terreno aplicando criterios basados en la curvatura local del campo escalar de alturas definido por el MDE, la posición del observador y el tamaño del triángulo. El grado de detalle se determina imponiendo una cota superior a la acumulación de un indicador extensivo (i.e. proporcional a la superficie) de curvatura corregido con la distancia al observador y el tamaño aceptable de los triángulos. El criterio es que las regiones con más curvatura y más cercanas al

observador requieren mayor nivel de detalle. De esta forma los lugares con baja curvatura pueden ser simplificados sin perder calidad [4].

El terreno o campo de alturas se transforma en una matriz cuadrada de dimensión $2^k x 2^k$, donde k es el máximo nivel de subdivisión permitido. El proceso comienza con los dos triángulos iniciales representantes del terreno y calcula para cada triángulo la curvatura local de sus cotas. Tales triángulos corresponden al hijo izquierdo y al derecho de un árbol binario que se usará para el almacenamiento de la triangulación. El indicador de error basado en la curvatura local establece hasta cuándo es necesario continuar con el refinamiento de cada región. Finalmente, los nodos terminales o frontera del árbol conforman la malla poligonal que se está visualizando en un dado instante (ver figura 1).

Otra problemática es la conformidad de la malla poligonal a nivel intrazonal e interzonal; es decir, evitar la presencia de roturas en la trama de la malla provocados por triángulos adyacentes con nivel de subdivisión disímiles o la articulación zonas adyacentes que provienen de distintos MDE. Para lograr una triangulación conforme, la jerarquía binaria no necesita restringirse por niveles como es el caso de los *quadtree* [7], simplemente recurre al tratamiento de cadenas de dependencias entre vértices. Permite generar alternativas de triangulación más flexibles, en las que se puede refinar localmente un triángulo sin afectar a los triángulos adyacentes. Esta problemática se soluciona incorporando una cota ficticia en la mitad del lado más largo del triángulo, su magnitud se calcula interpolando las cotas u extremos del lado. Las cotas ficticias se transforman en cotas reales como se indicará en la sección 3.

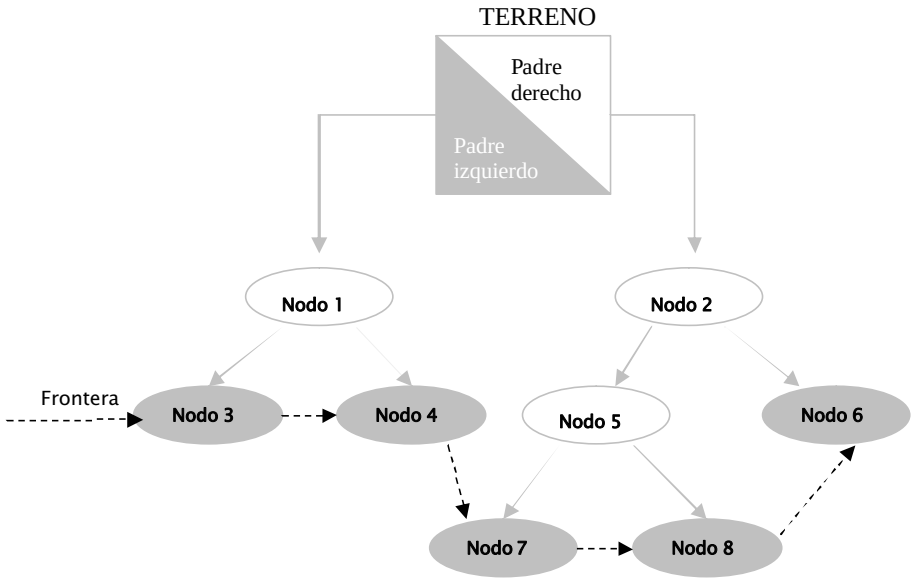


Figura 1- Representación jerárquica del terreno.

El algoritmo de simplificación comprueba en *O(n log n)* si el indicador de error sugerido se verifica en los nodos terminales (o frontera). Si el nodo cumple con las condiciones impuestas en el indicador de error, inmediatamente se procede a insertar el triángulo en una frontera auxiliar que reemplazará a la actual, en caso contrario el proceso subdivide al triángulo en dos triángulos (sus descendientes), incrementa en uno la cantidad de triángulos a representar, los inserta en la frontera auxiliar y repite el mismo proceso con cada uno de ellos.

Por otra parte, al desplazarse el observador, sucede que las zonas antes cercanas ya no necesitan el detalle que conservan y pueden remallarse [11]. Este proceso recursivo se inicia en los nodos terminales (frontera) y continúa con los padres, ascendiendo en el árbol hasta alcanzar el nivel de detalle deseado.

La selección automática de detalle se logra mediante la conjugación de tres factores relevantes que otorgan la información necesaria desde tres ópticas fundamentales: homogeneidad de zonas (factor de curvatura), tamaño válido del triángulo (factor tamaño) y posición relativa del observador con respecto a la zona (factor distancia). Con respecto a la homogeneidad, se evalúa la curvatura local de cada triángulo y se compara con un umbral derivado del cálculo según la ubicación del observador. Este criterio compara la desviación de la geometría intrínseca de la superficie respecto a la geometría del plano. Con respecto al tamaño del triángulo, conviene establecerlo como una medida proporcionada con la longitud de la diagonal del rectángulo que representa al terreno en análisis. Por último, el factor distancia resultará una medida relativa al valor de la distancia máxima.

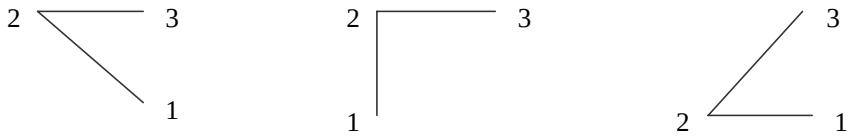
En las siguientes abstracciones procedurales se pude visualizar cómo funciona la selección automática de detalle, donde *triangular* es una función recursiva que incrementa el nivel de detalle en una zona.

```
agregar_detalle (vértice * observador) {

    para todas las zonas en la frontera
        si cercanía_al_observador(zona, observador) and
           tamaño_triángulo_válido( zona ) and
           no planaridad( zona, observador) entonces
              triangular(zona_hija_izquierda , observador)
              triangular(zona_hija_derecha , observador)
        sino
              actualizar frontera (zona);
}


sacar_detalle (matriz * zona , vértice * observador) {

        si cercanía_al_observador(zona , observador) or
           tamaño_triángulo_válido( zona ) or
           no planaridad( zona , observador) entonces

              restaurar_strip(zona);
              sacar_detalle(zona , observador);

}
```

En todos los algoritmos planteados se hace referencia al tamaño del triángulo, lo que nos da una referencia que indica cuán lejos estamos del tamaño límite impuesto. Finalmente, la distancia del observador a la zona debe ser aceptable para proporcionar una cuota de realidad dada por el alcance de una vista ficticia.

3 IMPLEMENTACIÓN

Los vértices de los triángulos se organizan en un *triangle strip* debido a que son construcciones eficientemente soportadas por el hardware gráfico y ocupan menos espacio que otras representaciones. Los triángulos surgen de tomar grupos de tres vértices contiguos del triangle strip, formando tres tipos de triángulos A, B y C (ver figura 2). Este agrupamiento genera triángulos línea, que hemos eliminado verificando la colinealidad de los vértices previamente.



Tipo A **Tipo B** **Tipo C**

Figura 2- Tipos de triángulos A, B y C. Subdivisión de un triángulo de tipo B.

La subdivisión de un triángulo (o secuencia de vértices) de tipo A genera dos nuevos triángulos o secuencias de tipos B y C respectivamente. Similarmente, cuando el triángulo es de tipo B, se generan dos triángulos de tipo A y C. La figura 3 muestra un triángulo de tipo B cuya secuencia generadora es {1, 2, 3}, en un paso intermedio se agrega a la secuencia el segundo vértice quedando {1, 2, 3, 2, 3}; por último, se genera la secuencia generadora de los dos triángulos hijos {1, 2, 4, 2, 3}. Cada secuencia determina el strip de un triángulo o una sucesión de triángulos. Finalmente, si el triángulo es de tipo C, se subdivide en dos subsecuencias de tipos A y B [1].

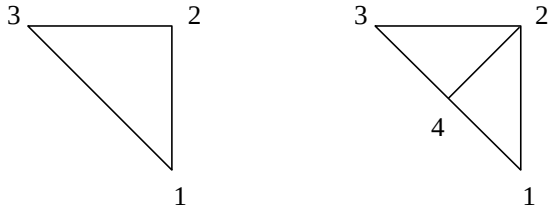


Figura 3 – Subdivisión de un triángulo de tipo B.

Los nodos del árbol conservan el número de triángulo por el cual se ordena la estructura que representa a la frontera, el tipo de triángulo (tipo A, B o C), y los punteros al *triangle strip*. También se conservan las coordenadas del vértice, su altura y un *estado* valuado en *seleccionado (S)* cuando el vértice posee la cota real, *ficticio (F)* cuando la cota surge de la interpolación de los extremos del segmento y *no seleccionado (D)* para desmarcar vértices, útil al proceso de compresión. Los algoritmos de compresión y descompresión restauran el *triangle strip*, decrementando o incrementando la longitud de la secuencia de vértices debido a que la cantidad de triángulos disminuye o se agrega.

Cuando se está por subdividir el triángulo, es muy importante fijarse qué va a pasar con la zona enfrentada en pos de evitar cracks y aproximar con realismo el terreno. Inicialmente, el *estado* del vértice 4 presente en la figura 4 es *ficticio*, por lo tanto su altura proviene de la interpolación. No obstante, como es compartido por triángulos adyacentes que han sufrido diferentes niveles de subdivisión, tendrá asociado un vértice dependiente: el vértice 6. Cuando su vecino adyacente

también se subdivida, el *estado* del vértice 4 será el del *seleccionado*, su valor tendrá el de la cota real y no tendrá dependiente, ya que el crack ha sido salvado en ambas zonas vecinas. De esta forma, mediante lógica de posibles estados del vértice, se evita rápidamente la formación de cracks, impidiendo el manejo de cadenas de dependencias recursivas entre zonas vecinas.

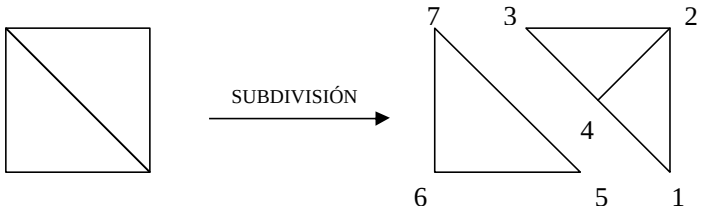


Figura 4 – Dependencias de vértices para eliminación de cracks

4 ANÁLISIS DE RESULTADOS

El algoritmo propuesto se aplicó a la región geográfica demarcada en la Figura 5, precisamente la franja terrestre definida entre los 67º y los 71º Oeste correspondiente a los 35º Sur. Tal información fue obtenida a partir de cinco archivos HGT disponibles en www.goldensoftware.com que representan a una región compuesta por 14,4 millones de triángulos. En este formato, cada archivo cubre áreas de 1º x 1º con resolución adecuada, definidos por 1201x1201 celdas o 2,88 millones de triángulos.

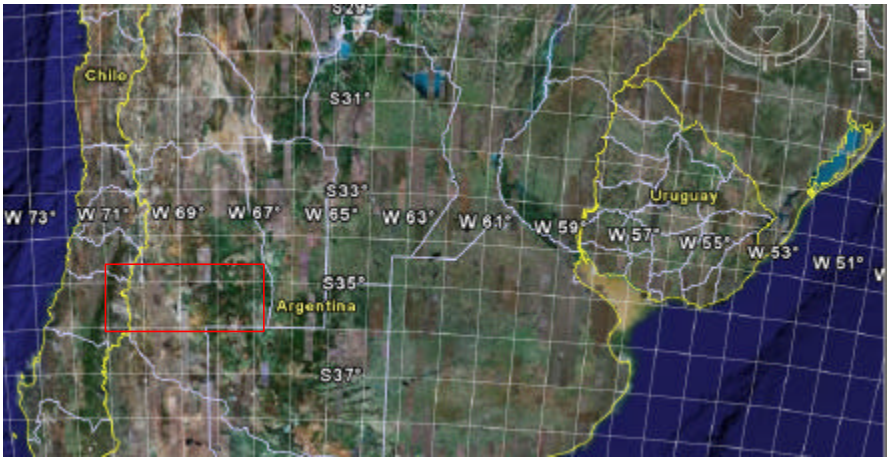


Figura 5 – Demarcación de la región geográfica de interés.

A pesar de los importantes cambios de pendiente en toda su extensión, se han logrado excelentes visualizaciones con simplificaciones con menos del 0,0004 % de los polígonos originales. La Figura 6 muestra seis aproximaciones de la región capturadas con una aplicación visualizadora que se desarrolló íntegramente en C++, cuya interfaz de usuario aprovecha las utilidades de la librería OpenGL [10]. La métrica de calidad de malla utilizada es la suma de las diferencias de cuadrados entre pares de cotas pertenecientes a la aproximación y la malla original cruda.

En principio, se unen los archivos HGT que definen la región de interés y luego comienza el proceso de simplificación poligonal del que resultan las aproximaciones con 10, 45, 845, 3549,

5873 y 5127 triángulos de la Figura 6. El indicador de error establece el nivel de detalle necesario en la región inicializando los tres factores que componen el criterio. A saber, el factor de tamaño del triángulo se inicializó como el 1% de la diagonal del rectángulo que representa al terreno original, el factor distancia corresponde a tomar el 80% del valor de distancia máxima y, finalmente, el factor asociado a la curvatura local queda establecido como el 20% del valor de curvatura máxima.

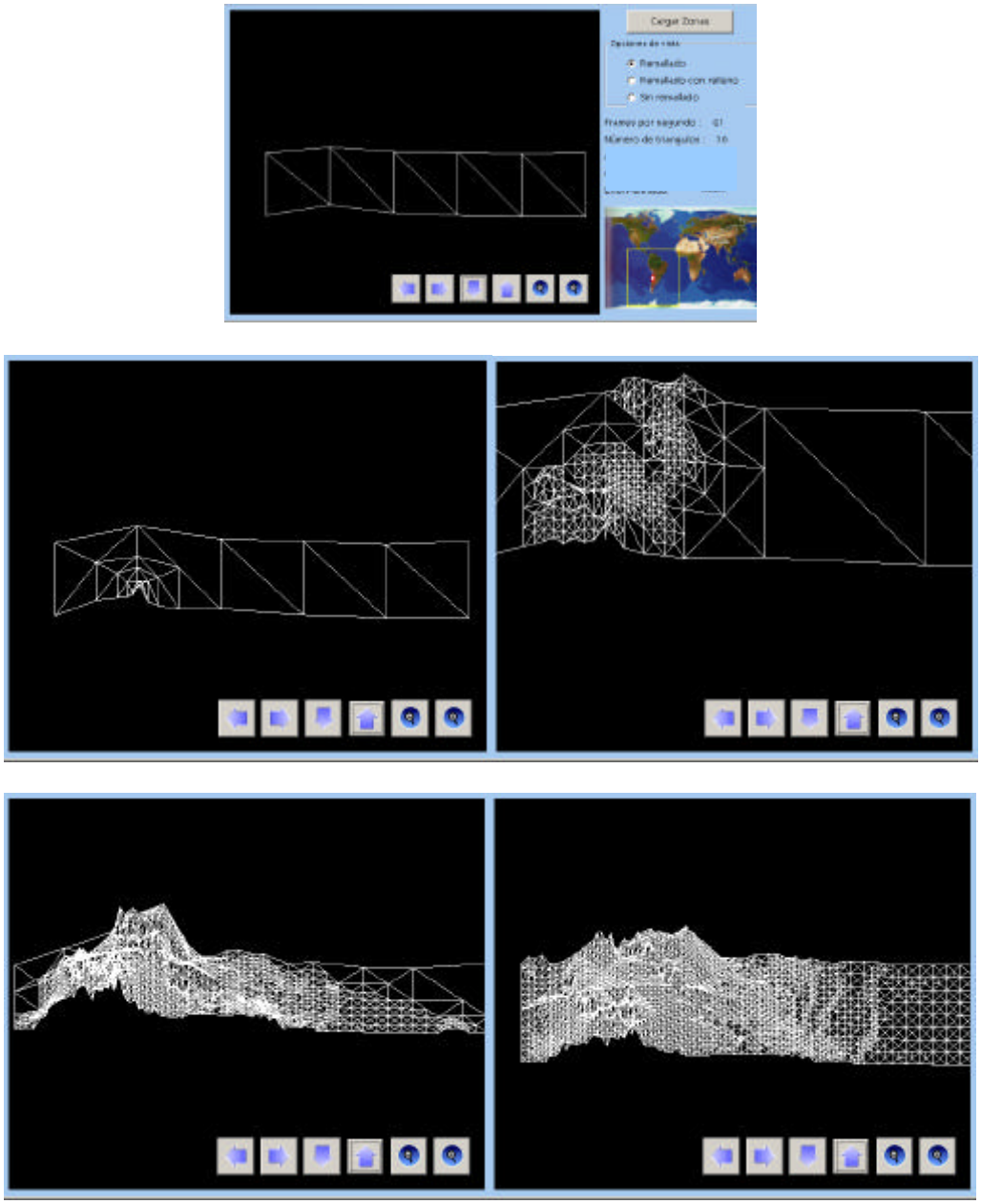


Figura 6 – Simplificación de topografía

5 CONCLUSIONES

Se presentó un algoritmo con una complejidad computacional de O(n log n) para la simplificación y visualización interactiva de modelos topográficos, con la cual se genera una jerarquía binaria de mallas poligonales del mismo terreno con distinta resolución. La selección automática de detalle se logra mediante un indicador que combina la posición del observador, el tamaño de los triángulos, sumado a un criterio de simplificación que es la curvatura local. Para evitar los cracks entre triángulos adyacentes, se agregan vértices ficticios que mantienen una cadena de dependencia con el vértice opuesto al triángulo analizado. Además se le suma una alternativa a la generación sistemática de triangle strip. El algoritmo de remallado mantiene la calidad visual tolerada por el usuario y detalla aún más las regiones dentro del campo de visión directo del observador. El algoritmo presentado simplifica notoriamente el volumen de datos a visualizar respecto de la grilla regular original.

REFERENCIAS

[1] Abásolo, M.J., Blat, J., De Giusti, A., A Hierarchical Triangulation for Multiresolution Terrain Models. *Journal of Computer Science and Technology, v.1, n.3, Computers Science Research: State of the Art,* 2000.

[2] Balmelli, L., Kovacevic, J., Vetterli, M., Quadtrees for embedded surface visualization: constraints and efficient data structures. *In Proceedings of IEEE International Conference on Image Processing,* 2:487-491, 1999.

[3] Cifuentes, M.V., D'Amato, J.P., García Bauza, C., Vénere, M.J., Clausse, A., Remallado interactivo como método para la navegación eficiente de modelos topográficos. *Mecánica Computacional*, 24:2385-2395, 2005.

[4] Gross, M.H., Gatti, R., Staadt, O., Fast Multiresolution Surface, Meshing. *Proceedings of Visualization'95,* p. 135-142, 1995.

[5] Hoppe, H., View-Dependent Refinement of Progressive Meshes. *Computers & Graphics. Proceedings of SIGGRAPH´97,* p.189-198, 1997.

[6] Lindstrom, P., Pascucci, V., Visualization of Large Terrains Made Easy. *Proceedings of IEEE Visualization, San Diego, California* , p. 363-370, 2001.

[7] Pajarola, R., Overview of quadtree-based terrain triangulation and visualization. *Technical Report, UCI-ICS-02-01, Information & Computer Science, University of California Irvine, ical Methods in Engineering*, 2002.

[8] Rottger, S., Heidrich, W., Slusallek, P., and Seidel, H. P., Real-Time Generation of Continuous Levels of Detail for Height Fields. *V. Skala, editor, Proceedings of of the 6th International Conference in Central Europe on Computer Graphics and Visualization, pages 315-322*, 1998.

[9] Vénere, M.J., Cifuentes, M.V., D'Amato, J.P., García Bauza, C., Editor de escenarios para aplicaciones de Realidad Virtual. *34º JAIIO, Jornadas Argentinas de Informática e Investigación Operativa*, 2005.

[10] Woo, M., Neider, J., Davis, T., OpenGL Programming Guide: The Official Guide to Learning OpenGL. *Addison-Wesley Developers Press*, 1997.

[11] Xu Qing, Zhang Baoming, Tan Bing, Ma Dongyang, A Dynamic Multi-resolution Model and It´s Application to Terrain Rendering. *Symposium on Geospatial Theory, Processing and Application. Ottawa, 2002.*

Scatter Plot 3D: SPG

Maximiliano J. Escudero
M. Luján Ganuza
Dyanna Wilberger
Sergio R. Martig
Departamento de Ciencias e Ingeniería de la Computación
VyGLab
Laboratorio de Investigación en Visualización y Computación Gráfica
Universidad Nacional del Sur
Avenida Alem 1253
Argentina, CP 8000, Bahía Blanca, Buenos Aires
{mje, mlg, srm}@cs.uns.edu.ar

Resumen

Debido a los grandes volúmenes de información generados en diversos ámbitos y en forma continua, resulta indispensable desarrollar técnicas para facilitar la tarea de quien debe utilizar dicha información. De la necesidad de estudiar y obtener técnicas efectivas para comunicar información a través de imágenes nace la Visualización. Una herramienta ampliamente utilizada para estudiar las relaciones entre los diferentes atributos de los items presentes en un conjunto de datos es el Scatter Plot. Un Scatter Plot 2D consiste en una representación gráfica utilizada en estadística para mostrar y comparar conjuntos de datos. Resulta útil diseñar una técnica de Scatter Plot 3D capaz de representar una mayor cantidad de atributos por item de dato haciendo, por un lado, uso de la tercera dimensión para mapear un valor al tercer eje y enfatizando la explotación de las propiedades gráficas del elemento de representación utilizado. Se propone entonces el uso de Glifos como elemento de representación en un Scatter Plot 3D, aumentando potencialmente la dimensionalidad del mapeo visual.

Keywords: Visualización de información, Scatter Plot, Glifos

1. INTRODUCCIÓN

Si bien el objetivo básico detrás del concepto de Visualización es la obtención de representaciones visuales interactivas con el propósito de la adquisición y el uso del conocimiento; según la naturaleza y características de la información a visualizar podemos hablar de tres tipos de visualización: Visualización Científica, Visualización de Software y Visualización de Información. La herramienta Scatter Plot 3D SPG presentada en este artículo encuadra en el último tipo de visualización, brindando la posibilidad de explorar relaciones entre los distintos items de datos.

2. SCATTER PLOT 3D: SPG

Entre las técnicas más populares de visualización se encuentra el Scatter Plot 2D, que grafica datos multidimensionales mapeando dos dimensiones a las coordenadas del plano XY, y mapeando otras dimensiones a los atributos gráficos de los elementos visuales que representan las tuplas de datos como color, tamaño, intensidad o forma [6]. Los Scatter Plots 3D corresponden a una extensión

conceptualmente simple de los Scatter Plot 2D [1] [7]. Esta técnica permite representar visualmente a tres o mas atributos de los items de datos. Si bien el mapeo espacial es perceptualmente dominante, es decir las relaciones existentes entre los atributos mapeados a las coordenadas espaciales se perciben con mayor claridad, se pueden representar atributos adicionales enriqueciendo el mapeo visual. Eventualmente se podrían agregar más coordenadas al modelo, llevándolo al espacio n-dimensional. Los Scatter Plots 3D se utilizan principalmente cuando resulta imposible disponer los datos a mostrar sobre una grilla rectangular de dos dimensiones, en cuyo caso un Scatter Plot 2D resultaría insuficiente. En versiones más complejas, pueden incluir atributos específicos para los elementos de representación según el dato representado (color, tamaño, orientación, forma, etc.), guías y combinaciones de datos scatter con objetos adicionales como superficies de ajuste.

En la figura 1 se puede observar un Scatter Plot en tres dimensiones que hace uso del tamaño variable de las esferas para mapear un atributo adicional.

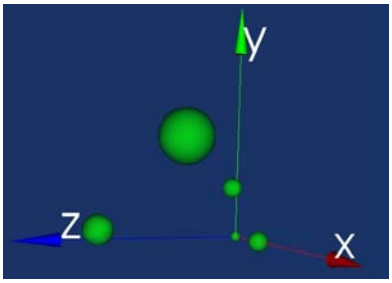


Figura 1: Visualización generada con el Scatter Plot 3D donde el elemento de representación es una esfera.

2.1. CARACTERÍSTICAS

Los Scatter Plots 3D presentan ciertas ventajas y limitaciones respecto a los modelos en 2D. A continuación se detallan algunas de ellas.

2.1.1. Ventajas

- En un Scatter Plot 3D mantener la misma densidad de puntos que en un Scatter Plot 2D implica aumentar el número de datos experimentales a mostrar (mayor espacio muestral). Si se mantiene el número de puntos del scatter plot inicial (2D), existe mayor descripción de las relaciones existentes entre variables, dado que se incorpora una característica más de los datos. En la figura 2 se puede observar el mismo conjunto de datos representado por un Scatter Plot en dos dimensiones y por uno en tres dimensiones. Es claro que en el segundo los datos se muestran más espaciados, esto depende, por supuesto, de cómo sean mapeados los atributos.

- Permite incrementar el número de atributos que se pueden visualizar simultaneamente. Agrega la tercera dimensión lo cuál ya agrega una dimensión al mapeo visual, la coordenada Z, y permite mapear otros atributos a características extras como, por ejemplo, el el tamaño del objeto.

- El uso de Scatter Plots 3D con glifos para representar los datos brinda la posibilidad de utilizar técnicas de generación procedural de formas [5]. Estas técnicas permiten aumentar el número de dimensiones de los datos a mostrar explotando la forma de los glifos, aprovechado así la habilidad pre-atentiva del sistema visual humano para discriminar formas.

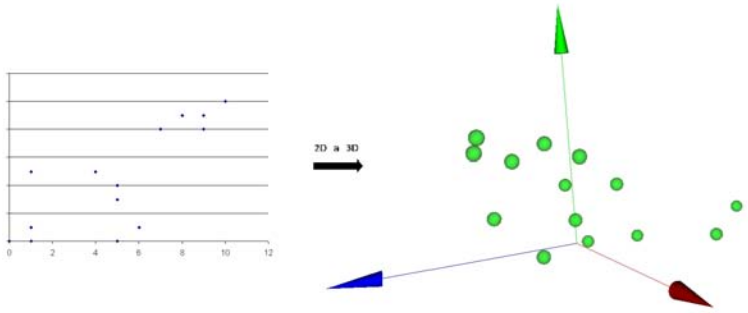


Figura 2: Representación de un conjunto de datos en 2D y en 3D

- Si se logra un mapeo de atributos eficiente, y se proveen las interacciones necesarias para la navegación y consulta, el poder expresivo de un Scatter Plot 3D puede asistir en el análisis de relaciones complejas entre múltiples variables.

2.1.2. Limitaciones

- A medida que aumenta la dimensionalidad se incrementa de forma exponencial el número de datos experimentales necesarios para poder estimar de manera exacta una función de aproximación. En la mayoría de los casos, la información que se pierde por considerar alguna característica extra de los datos, mapeada en la tercera dimensión, se compensa por un mapeo más exacto en un espacio de menor dimensión. En otras palabras, no es recomendable abusar de la multidimensionalidad si no es absolutamente necesario y el resultado no es visualmente ilustrativo.

- Trasladar representaciones de información en 2-dimensiones a 3-dimensiones no es una tarea simple, ya que la dimensión extra afecta enormemente el modo en que la información puede ser presentada. Es importante considerar que la visualización debe realizar un uso eficiente de la nueva dimensión. Además, se debe evitar que la nueva representación sea mal interpretada por el usuario como consecuencia de un mapeo inapropiado.

- Es necesario tener especial consideración respecto de la percepción de la distancia espacial. El tamaño de los objetos puede provocar que el usuario no perciba la perspectiva correcta de la información mostrada; es decir, se dificulta la discriminación entre las distintas profundidades de los objetos.

- Una desventaja proveniente del uso de objetos tridimensionales es la oclusión, que se da cuando un objeto tapa a otro ocultándolo en el espacio muestral. Este tipo de problema se presenta principalmente cuando la densidad de items de datos a mostrar es grande, o cuando simplemente un objeto muy grande se posiciona por delante de objetos más pequeños. Este tipo de efecto no es exclusivo del espacio 3D sino que está presente también en entornos 2D.

- Si las múltiples dimensiones mostradas en un volumen 3D pueden ser separadas en 2 o más dimensiones 1D o 2D de información ortogonal no existe beneficio alguno en el uso de una visión 3D. Esto es, los atributos de información son independientes unos de otros; lo que conlleva a que el usuario no descubra ninguna nueva relación a partir de una visión integrada.

2.2. Glifos como elementos de representación.

En base a todo lo expuesto hasta aquí, se ha diseñado e implementado un Scatter Plot 3D basado en glifos, al que llamaremos “SPG”.

Los glifos consisten en objetos gráficos en 3 dimensiones que representan uno o más valores en una única locación del espacio. Son formas o imágenes generadas por mapeo de los componentes de la tupla de datos a sus atributos gráficos. Un único glifo puede utilizarse para representar muchas propiedades o atributos de los datos en un punto dado del espacio, de forma tal que resultan ser herramientas muy útiles a la hora de interpretar una gran cantidad de información a primera vista.

Existen muchos tipos de glifos, generados a partir de distintas figuras geométricas. El SPG se basa en un glifo formado por una esfera semi-transparente en su centro y una cantidad variable de picos sobre la superficie de esa esfera. Además se debe tener en cuenta que la altura de dichos picos puede variar como así también su color. Este glifo se puede observar en la figura 3.

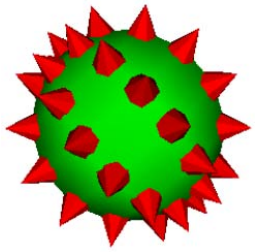


Figura 3: glifo SPG

2.3. Implementación del SPG

El scatter plot “SPG” es una aplicación que toma un conjunto de datos de entrada que pueden residir en un archivo o ser creados proceduralmente y genera una visualización en 3D basada en glifos [Figura 4], donde por aca tupla se genera un glifo y se posiciona en un determinado punto del espacio de mostrado.

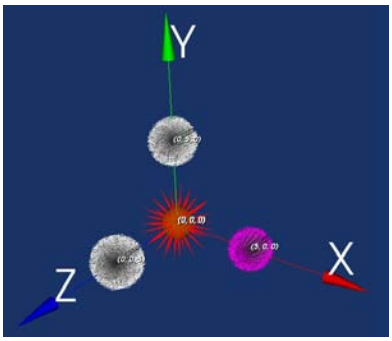


Figura 4: Visualización SPG

2.3.1. Librería VTK

Para implementar el Scatter Plot 3D con glifos se utilizó la librería VTK 5, la cual es una librería implementada y diseñada bajo los conceptos de la programación orientada a objetos. La misma

está compuesta por librerías de C++ y una capa interpretada que permite manipular las clases compiladas usando Java, Tcl y Python.

Una aplicación básica en VTK consiste en dos partes: la construcción del pipe para procesar los datos y la creación de los objetos gráficos para mostrarlos. Uno de los objetos básicos del modelo gráfico de VTK es el vtkActor, cuyas instancias representan objetos en la escena. En particular, vtkActor combina las propiedades del objeto (color, tipo de sombreados, etc.), la definición de la geometría y la orientación en el sistema de coordenadas del mundo.

Al implementar el SPG utilizando vtkActor para representar los distintos objetos involucrados en las escenas generadas, notamos una muy baja performance al intentar visualizar conjuntos de datos grandes (de más de 100 nodos), es por esto que se optó por la utilización de la clase ***vtkLODActor,*** que se comporta de manera similar a vtkActor con la única diferencia de que genera objetos a mucha menor resolución, lo que induce a visualizaciones menos atractivas visualmente pero mucho mas manipulables a la hora de aplicar interacciones que involucren la exploración de la escena.

2.3.2. Mapeo de Atributos Gráficos

En una primera instancia, el SPG permite mapear atributos de los items de dato a 6 atributos gráficos del glifo, con posibilidad de extender el mapeo a 8 atributos si se agregan las interacciones necesarias.

Los 6 atributos gráficos básicos son los siguientes:

- Posición del glifo (coordenadas x, y, z): Se pueden mapear atributos escalares a las coordenadas x, y, z de la escena, definiendo la posición del glifo.
- Radio de la Esfera: Se puede mapear un atributo al radio de la esfera, definiendo su tamaño. La aplicación SPG se encarga de normalizar los datos mapeados a este atributo gráfico, de forma tal que todas las esferas quedarán con un tamaño de entre 1 y 50.
- Color de los Conos: Se puede mapear un atributo al color de los conos. A nivel del prototipo se restringe la cantidad de colores posibles a una cantidad finita, impactanto la cardinalidad del conjunto de datos a mapear a este atributo sobre la calidad de mapeo visual.
- Cantidad de Conos: Por último, se puede mapear un atributo a la cantidad de los Conos del glifo. Estos datos también serán normalizados por la aplicación (procurando que no se intenten generar mas datos de los que soporte la misma, ya que si la cantidad de conos es muy grande la performance de la visualización se degrada demasiado). Estableciéndose entonces una cantidad máxima de conos (50) y una cantidad mínima (1) para asegurar que el glifo mantenga su forma.

Los primeros tres atributos dan la posición del glifo en las coordenadas de la escena. La aplicación SPG brinda también la posibilidad de visualizar esta posición en forma numérica mediante etiquetas que se posicionan por delante del glifo, como se observa en la figura 4. Eventualmente, estas etiquetas se podrían utilizar para mapear mas datos, pero debe tenerse sumo cuidado de no sobrecargar demasiado la vista, es por eso que en el SPG inicial no se da uso a esa facilidad.

La visualización muestra como referencia los ejes XYZ a partir de los cuales se posicionan los glifos. La longitud de estos ejes es variable, y se adapta al conjunto de datos mostrado. Es decir, si el conjunto de datos mostrado tiene todos los glifos posicionados cerca del origen, entonces la longitud de los ejes será pequeña; en cambio si los glifos se encuentran esparcidos se calcula la posición del más lejano al origen y en base a eso se determina la longitud apropiada de los ejes.

Como se mencionó al inicio de esta sección, es posible mapear dos atributos adicionales a los glifos, aumentando la cantidad de datos mapeados a atributos gráficos de 6 a 8. Esto es posible sólo

si se agregan las interacciones adecuadas sobre la visualización, que permitan revelar detalles que de otra forma no serían claramente perceptibles por el usuario.

Los posibles atributos a mapear serían:

- Longitud de los conos: Se puede utilizar la altura de los conos de los glifos para mapear un dato adicional.
- Color de las Esferas: Se podría mapear un atributo al color de las esferas, con las mismas restricciones que para el color de los conos.

Estos dos atributos gráficos no se encuentran dentro de los atributos gráficos básicos porque su mapeo introduce graves problemas. Al mapear un dato a la altura de los conos se obtiene una escena con glifos de altura variables, lo que provoca cierta dificultad a la hora de percibir el tamaño real de las esferas. En consecuencia se lograría mapear un dato más pero se perdería detalle sobre otro atributo. Es por esto que en caso que se desee hacer uso de este atributo gráfico es necesario incorporar la interacción ***"Mostrar/Ocultar Conos"***. De esta manera se podría distinguir entre las distintas longitudes de los conos cuando estos estén visibles, y ocultarlos para detectar los distintos tamaños de las esferas.

En la figura 5 se puede observar el efecto de esta interacción. En la figura de la derecha se muestra una visualización generada con el SPG en la cual la variabilidad de la longitud de los conos dificulta apreciar el tamaño de las esferas, que es el mismo para todos los glifos. Propiedad que si se observa en la figura de la izquierda.

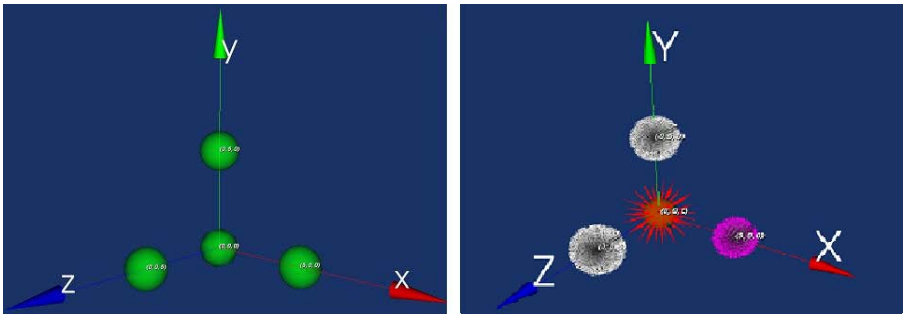


Figura 5: Visualización generada con el SPG con los conos ocultos (izquierda) y con los conos visibles (derecha).

Algo similar ocurre con el Color de las Esferas. Es claro que el usuario es incapaz de distinguir el color de las mismas si éstas se encuentran tapadas por los conos. Aquí también surge la necesidad de la interacción "Mostrar/Ocultar Conos", que permitirá hacer visible las esferas limpias distinguiendo así el color que se les ha asociado.

2.3.3. Interacciones

Además de la explicada en la sección anterior, el SPG brinda otras interacciones al usuario para lograr una exploración de los datos más amigable. A continuación se explican cinco posibles interacciones:

- ***Mostrar/Ocultar Conos:*** Consiste simplemente en hacer visibles u ocultar los conos de los glifos, dejando visibles (en caso de ocultarlos) sólo las esferas. La necesidad de contar con esta interacción fue explicada en la sección anterior.

- ***Mostrar/Ocultar Etiquetas:*** Se mencionó anteriormente que la aplicación SPG muestra mediante etiquetas la posición de cada uno de los glifos. Esta información puede provocar molestias cuando la cantidad de datos a mostrar es muy grande, los glifos se encuentran muy cerca espacialmente, o se hace un zoom out de la vista, generando una visualización poblada de etiquetas que taparán el detalle de lo que en realidad se desea mostrar. Se desprende entonces la necesidad de contar con una interacción que permita ocultar estas etiquetas, y hacerlas visibles cuando el usuario así lo requiera. En la figura 6 se puede observar el efecto de aplicar esta interacción a una visualización generada por el SPG sobre un conjunto de datos de 80 nodos. Es claro que la imagen de la izquierda (donde las etiquetas están ocultas) resulta mucho más ilustrativa que la de la derecha.

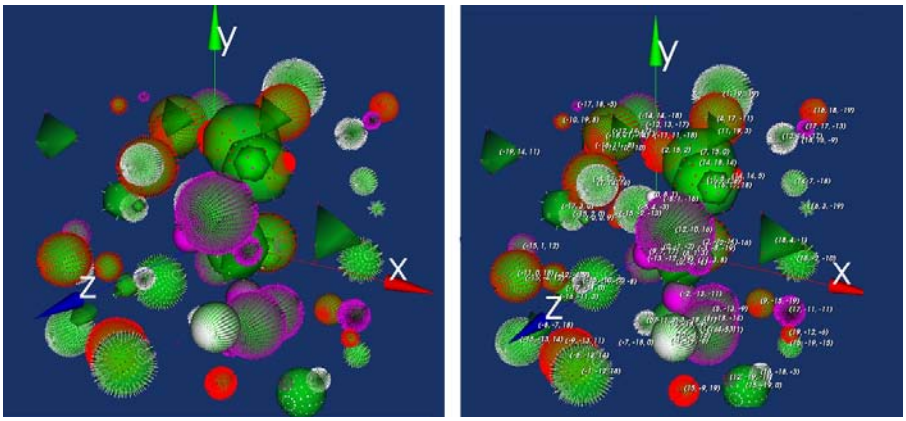


Figura 6: Visualización generada con el SPG con etiquetas ocultas (izquierda) y con los etiquetas visibles (derecha).

- ***Etiqueta Bajo Demanda:*** Como alternativa a la interacción anterior el usuario puede solicitar ver una etiqueta correspondiente a un único nodo. Esta interacción se puede observar en la figura 7.

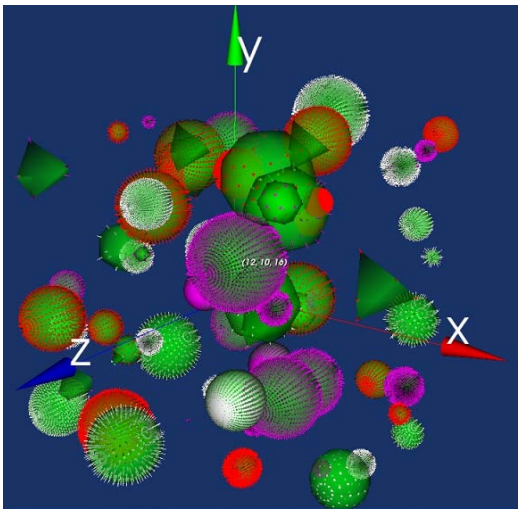


Figura 7: “Interacción Etiqueta bajo demanda”.

- ***Agregar Identificadores:*** Esta interacción permite al usuario asignar identificadores a los atributos gráficos, para facilitar otro tipo de interacciones. Por ejemplo interacciones de filtrado, donde el usuario deberá indicar sobre que atributo se desea efectúar el filtrado, y para esto necesita una forma de nombrarlos dependiente de lo que está mostrando. Por ejemplo, si se está visualizando una base de datos de un conjunto de asociados a una Obra Social, y el color de los conos representa el sexo de los asociados, entonces el usuario podría mediante esta interacción registrar esta relación en la vista.
- ***Zoom Semántico:*** Esta es una interacción un poco más compleja que las anteriores, mediante la cual se brinda detalle bajo demanda. Estos datos no formarán parte de los glifos, sino que se presentarán en algún formato alternativo. Por ejemplo: al hacer click sobre un glifo se abrirá una ventana que muestre las propiedades del item de dato asociado a ese glifo que no han sido mapeadas a atributos gráficos.
- ***Explotar Glifos:*** Es posible agrupar visualmente varios datos que presenten cierta similitud en un único glifo, el usuario podrá en ese caso seleccionar ese glifo y mediante esta interacción "explotarlo" para ver los datos agrupados en el mismo.

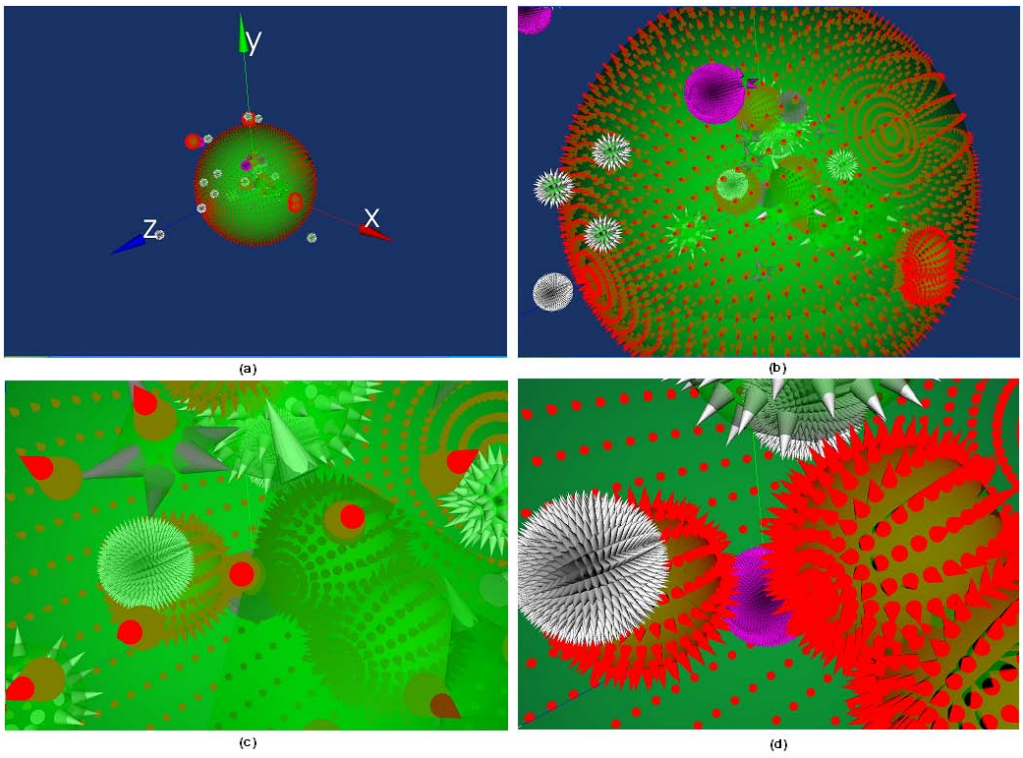


Figura 8: Secuencia de imágenes que ilustran la interacción "Explotar un Glifo".

2.3.4. El Problema de la Oclusión

Durante el desarrollo e implementación de la aplicación SPG se detectó un problema importante, que se daba en caso que dos o mas items de datos fueran mapeados a glifos situados en locaciones espaciales muy cercanas. Esto daba lugar a visualizaciones donde algunos glifos quedaban ocultos detrás de otros, dado su inferior tamaño o su ubicación en la pantalla relativa al usuario. Se detectaron tres situaciones posibles:

- Glifos Contenedores: Son aquellos cuyas esferas son tan grandes que contienen en su interior otros glifos más pequeños.
- Glifos Obstruyentes: Son aquellos que se ubican por delante de otros, tapando parcialmente o totalmente los glifos situados por detrás.
- Glifos Coincidentes: Son aquellos cuyas coordenadas espaciales y radio de la esfera coinciden, de forma tal que resulta imposible identificar la existencia de más de un glifo en un mismo punto.

El problema que surgió a partir de la presencia de los Glifos Contenedores y de los Glifos Obstruyentes se solucionó permitiendo cierto grado de transparencia sobre la superficie de las esferas, brindando al usuario la posibilidad de discriminar que hay más allá de la superficie de las esferas que se encuentran más adelante (desde el punto de vista el usuario). La figura 8 ilustra esta característica mostrando una secuencia de 4 imágenes correspondientes a una visualización generada en SPG a partir de un conjunto de datos de 40 nodos. En la figura 8 (a) se muestra un glifo compuesto por un conjunto de otros glifos. En la figura (b) se ilustra un instante en el cual el usuario se acerca al glifo, en la figura (c) se está atravesando la superficie del glifo contenedor, y finalmente en la última imagen se puede observar el detalle de los datos contenidos por el mismo.

El problema de los Glifos Coincidentes aún no ha sido resuelto. Una posible solución consiste en hacer uso de las facilidades provistas por las etiquetas, agregando una etiqueta especial (solo visible sobre glifos coincidentes) que indique la cantidad de glifos situados sobre el mismo punto. Los datos correspondientes a los glifos situados en ese punto serían representados por un único glifo cuyas características estén gobernadas por un promedio de los valores asociados a cada uno de atributos gráficos de los Glifos Coincidentes individuales.

3. CASO DE ESTUDIO: BD DE INGRESANTES A LA UNS

Debido a que, con grandes volúmenes de información resulta demasiado complejo hacer inferencias sobre qué es lo que está ocurriendo con los datos, nuestro objetivo en este caso de estudio es descubrir los patrones presentes en el espacio de información subyacente de manera rápida y sencilla.

Para analizar el comportamiento del Scatter Plot SPG se han efectuado diferentes ensayos sobre una muestra de datos correspondientes a ingresantes a la UNS. Dichos datos están vinculados a la información de los alumnos que ingresan a la universidad en diferentes años a los cuales se les toma un examen de carácter nivelatorio de acuerdo a la carrera.

Cada ítem del conjunto de datos corresponde a un alumno, y esta compuesto por los siguientes atributos:

- Edad del Alumno.
- Sexo del Alumno.
- Número de Libreta del Alumno.
- Año de inscripción a la UNS.
- Nota obtenida en el examen nivelatorio correspondiente.

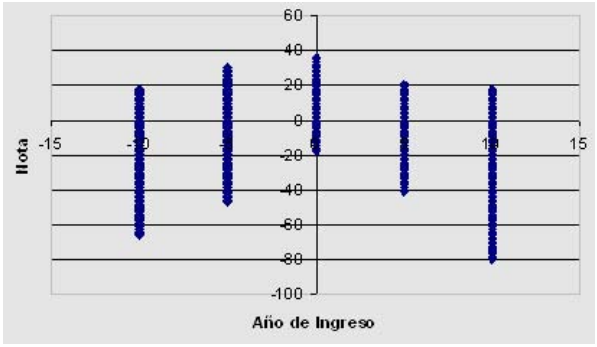
Atributo del Dato	Atributo Gráfico
Edad	Tamaño de la Esfera
Sexo	Color de la Esfera
Número de Libreta	Etiqueta
Año de Inscripción	Coordenada X
Nota Obtenida	Coordenada Y

(a) Mapeo 1

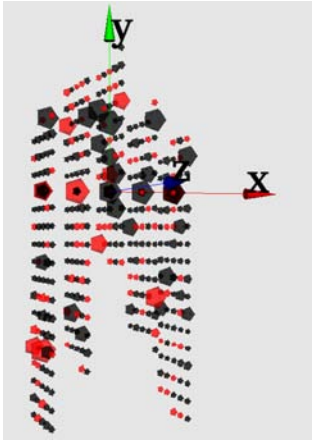
Atributo del Dato	Atributo Gráfico
Edad	Coordenada Y
Sexo	Coordenada Y
Número de Libreta	Etiqueta
Año de Inscripción	Tamaño de la Esfera
Nota Obtenida	Color de la Esfera

(b) Mapeo 2

Figura 9: Cuadros de Mapeos.



(a) Mapeo en 2D



(b) Mapeo en 3D

Figura 10: Comparación entre Visualizaciones generadas en 2D y 3D.

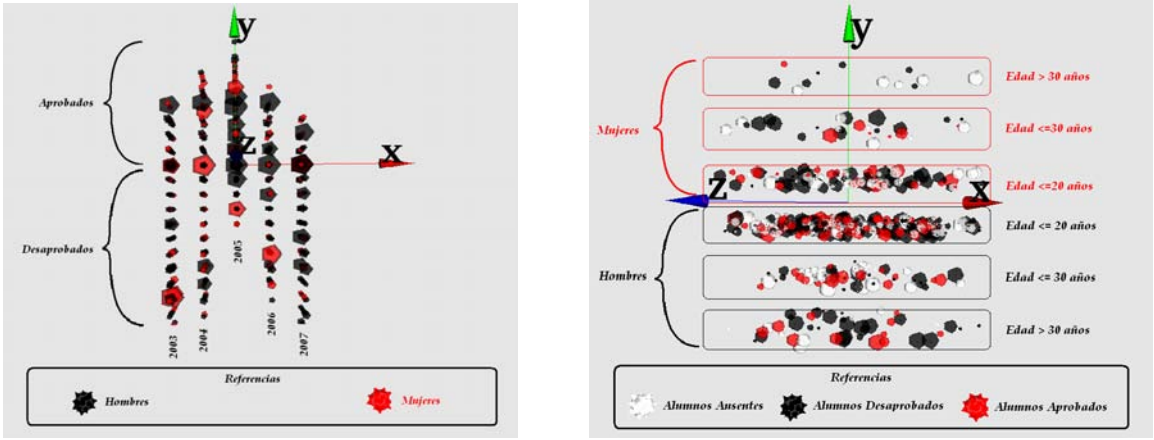
A la hora de representar un ítem de dato mediante atributos gráficos una visualización en dos dimensiones resultaría insuficiente, ya que solo permitiría mapear como máximo tres datos a atributos gráficos, de donde se desprende la necesidad de expandir la herramienta a tres dimensiones.

Con estos datos, se efectuaron diferentes mapeos a atributos gráficos con el fin de observar y comparar los resultados obtenidos, dichos mapeos se detallan en la figura 9.

Mapeo 1 El primer mapeo muestra los alumnos en 5 planos paralelos, cada uno representando un año de ingreso diferente. La ubicación de cada Glifo depende de la nota obtenida por el alumno representado. Si el Glifo se encuentra ubicado en una posición con valor de coordenada **y** mayor que cero corresponde a un alumno aprobado, si se encuentra ubicado en una posición con valor de coordenada **y** menor que cero corresponde a un alumno desaprobado, y en caso contrario (valor de coordenada y igual a cero) se trata de un alumno ausente. En la figura 11(a) se detalla la visualización obtenida mediante este mapeo.

Es claro que a primera vista esta visualización permite observar la diferencia entre la cantidad de alumnos aprobados y desaprobados por año, como así también permite la comparación entre años. Por ejemplo, de la figura 10(b) se puede inferir que en el año 2005 los alumnos presentaron un mejor desempeño que en los restantes años ya que por única vez la cantidad de alumnos aprobados es mayor a la de desaprobados.

A pesar de que este mapeo genera una visualización aceptable para el reconocimiento de patrones sobre el conjunto de datos, presenta también ciertas desventajas, como por ejemplo, que la posición de los alumnos dependa de la nota induce que el ángulo de visión influya sobre el entendimiento de la capacidad de visualizar correctamente los alumnos aprobados y desaprobados



(a) Mapeo 1 (b) Mapeo 2

Figura 11: Visualización generada con el SPG para caso de estudio con referencias.

de cada año.

Para recalcar los beneficios de la tercera dimensión en la figura 10(a) se muestra este mapeo proyectado en dos dimensiones. Observar que a pesar que es posible visualizar la cantidad de aprobados y desaprobados por año, como así también efectuar una comparación entre los distintos años; no es posible representar los demás atributos de cada item de dato. Como la edad, el sexo, el número de libreta, etc.

Mapeo 2 El segundo mapeo con el cual se experimentó se detalla en la figura 11(b). Según esta distribución los items de datos se disponen en regiones paralelas al plano **xz**, donde cada región representa un rango edades de los alumnos. Además, todos los glifos ubicados en posiciones con valor de coordenada **y** mayor que cero representan mujeres, y los ubicados en posiciones con **y** menor que cero representan hombres. El tamaño de la esfera denota el año de ingreso y el color la nota obtenida. En la figura 9(b) se detalla la visualización obtenida mediante este mapeo.

Como la posición del glifo no depende de la nota obtenida el ángulo de visión no influye sobre la vista de alumnos aprobados y desaprobados, pero si influye al comparar o recalcar la edad de los mismos.

Esta visualización permite fácilmente efectuar comparaciones entre el sexo de los alumnos y la edad, aunque dificulta apreciar la nota obtenida.

De estos dos mapeos se deduce claramente que la forma en que se elija mapear los datos sobre atributos gráficos influye directamente sobre el insight del usuario sobre la visualización resultante.

4. CONCLUSIONES

Se ha desarrollado una aplicación de Visualización que implementa la técnica Scatter Plot en tres dimensiones explotando la capacidad de representación de los Glifos.

La aplicación SPG permite una amplia flexibilidad respecto al conjunto de datos a representar, ya que permite el mapeo potencial de 8 datos a atributos gráficos. Adicionalmente provee interacciones que facilitan la exploración, consulta y navegación sobre el conjunto de datos, generando visualizaciones ilustrativas que permiten al usuario lograr insight sobre los patrones que presentan esos datos.

El uso potencial del SPG se extiende a todas las áreas de visualización presentadas en el capítulo uno, siempre y cuando los datos a mapear se ajusten a las restricciones establecidas en el capítulo tres.

5. AGRADECIMIENTOS

El presente trabajo fue parcialmente financiado por PGI 24/ZN12, Secretaría General de Ciencia y Tecnología, Universidad Nacional del Sur, Bahía Blanca, Argentina.

REFERENCIAS

[1] B. G. Becker. Volume rendering for relational data. pages 87–91.

[2] Benjamin B. Bederson and Ben Shneiderman. *The Craft of Information Visualization: Readings and Reflections*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[3] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[4] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A . Tukey. *Graphical Methods for Data Analysis*. Wadsworth Internat. Group, 1983.

[5] D. S. Ebert, R. M. Rohrer, C. D. Shaw, P. Panda, J. M. Kukla, and D. A. Roberts. Procedural shape generation for multi-dimensional data visualization. *Computers & Graphics*, 24:375–384, 2000.

[6] J. Fekete and C. Plaisant. Interactive information visualization of a million items proceedings of ieee symposium on information visualization, 2002.

[7] R. Kosara, G. Sahling, and H. Hauser. Linking scientific and information visualization with interactive 3d scatterplots, 2004.

[8] Sergio Martig, Silvia Castro, Pablo Fillottrani, and Elsa Estévez. Un modelo unificado de visualización. In *Proceedings 9º Congreso Argentino de Ciencias de la Computación.*, pages 881–892, 2003.

[9] P. Rheingans and C. Landreth. Perceptual principles for effective visualizations, 1995.

[10] Bernice E. Rogowitz, Lloyd A. Treinish, and Steve Bryson. How not to lie with visualization. *Comput. Phys.*, 10(3):268–273, 1996.

[11] Ware, Colin. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, 2000.

Seguimiento de Objetos en Video usando Contornos Activos y *Bounding Boxes*

Diego R. Park **María J. Gambini** **Marta E. Mejail**

Departamento de Computación, Universidad de Buenos Aires
Intendente Güiraldes 2160, Ciudad Universitaria
Ciudad de Buenos Aires, C1428EGA, Argentina

{dpark, jgambini, marta}@dc.uba.ar

Abstract

The automatic tracking of objects along a sequence of images has applications in different areas as robotics, animation, security systems or medical diagnosis. The tracking algorithm used in this paper starts fitting the contour of an object, using a B-Spline curve as the initial search region. The next step is to sample normal vectors at regularly-spaced points along this curve and to detect points on the border of the object by applying some image-processing filter along the curve normals. A good initial estimate is required for the tracking algorithm to be successful.

This paper presents a method to avoid parametrization errors when fitting the outline of the object at the beginning of the tracking. It has the advantage of being simple and efficient. Conflicts when fitting the contour of the object are avoided using an acceleration structure. The modified algorithm is tested against real videos with excellent results.

Keywords: Image Processing, Active Contours, Bounding Boxes, Video Tracking.

Resumen

El seguimiento de objetos en forma automática a lo largo de una secuencia de imágenes tiene aplicaciones en áreas tan diversas como robótica, animación, sistemas de seguridad o diagnóstico médico. El algoritmo de seguimiento utilizado en este trabajo comienza con la definición de una curva B-Spline que es el área inicial de búsqueda del contorno de un objeto. Luego se consideran una serie de segmentos de rectas normales a esta curva y se aplica algún método de detección de bordes para hallar puntos sobre el contorno a lo largo de las rectas. Para que el algoritmo de seguimiento del objeto sea exitoso es necesario que la estimación inicial sea muy precisa.

En este trabajo se presenta un nuevo método estable y eficiente para evitar errores de parametrización al ajustar el contorno del objeto con una curva B-Spline al comienzo del método de seguimiento. Se utiliza una estructura de aceleración para evitar conflictos al estimar el contorno del objeto. El algoritmo modificado se prueba en videos reales y se observan excelentes resultados.

Palabras clave: Procesamiento de Imágenes, Contornos Activos, *Bounding Boxes*, Seguimiento en Video.

1 INTRODUCCIÓN

El seguimiento de objetos en video es de gran utilidad en numerosas aplicaciones como robótica [16, 14], diagnóstico médico [6, 5, 20], monitoreo de sistemas de seguridad [7, 18], animación [28, 1, 22] y análisis de imágenes de Radar de Apertura Sintética (SAR) [11, 10], entre otros ejemplos.

Los algoritmos de contornos activos y *snakes* son muy utilizados para seguimiento de objetos en video, por su robustez y tratabilidad. El primer algoritmo de detección de bordes para imágenes ópticas, basado en evolución de curvas por medio de la minimización de la energía y cálculo variacional, fue desarrollado por Kass *et al.* [15]. Se trata de curvas que se deforman hasta ajustar el contorno de un objeto de interés.

En este trabajo se utiliza la representación de curvas B-Spline como herramienta fundamental en la descripción de contornos. El contorno formulado por medio de curvas B-Spline, tiene varias ventajas con respecto a otras representaciones de curvas porque permite control local, requiere pocos parámetros y es una función suave. Además de aplicarse al seguimiento de objetos en secuencias de video [3, 2, 17], la representación B-Spline se utiliza también para la aproximación de formas [26, 21, 8] y detección de bordes [4, 13].

Utilizamos aquí el algoritmo de seguimiento en secuencias de imágenes desarrollado por Blake *et al.* [2] que utiliza la representación B-Spline para describir curvas y el espacio de formas. El algoritmo comienza con una curva B-Spline inicial que ajusta al contorno del objeto en el primer cuadro de la secuencia. Para encontrar esta curva es necesario hallar los puntos de borde del objeto. Con este objetivo, se define un área inicial de búsqueda determinada también por una curva B-Spline y se consideran rectas equiespaciadas normales a esta curva. Luego se aplica algún algoritmo de detección de bordes sobre los segmentos de recta. Entonces la imagen se recorre por regiones en lugar de hacerlo sobre toda la imagen, lo que significa un gran ahorro en costo computacional. Luego, la curva inicial se deforma según movimientos permitidos, restringidos al espacio de formas. Sin embargo, una mala elección de normales al comienzo del algoritmo de seguimiento puede significar un ajuste poco adecuado, razón por la cual el algoritmo falla.

En este trabajo, presentamos una nueva solución para evitar los problemas que acarrea una mala elección de segmentos de recta normales a la curva. Se utilizan *Bounding Boxes* para obtener una curva B-Spline de ajuste muy preciso al comienzo del seguimiento.

Las *Bounding Boxes* se han utilizado en muchas aplicaciones, tanto para la detección aproximada de colisiones [24, 27, 29, 25] como estructura de aceleración [19, 9, 12]. Esta representación simplifica la geometría del objeto tratándolo como un rectángulo que lo contiene (de ahí *caja contenedora*).

Este trabajo está organizado de la siguiente manera: la sección 2 está dedicada a la representación de curvas B-Spline, al ajuste de puntos por medio de B-Splines y al espacio de formas. En la sección 3 se explica la aplicación de filtros a lo largo de normales para detectar los puntos de borde de un objeto y se discuten los problemas que surgen de una mala elección de normales. También se presenta el método propuesto que utiliza *Bounding Boxes* para solucionar el problema del cruce de normales, lo que constituye el aporte más importante de este trabajo.. En la sección 4 se muestran los resultados obtenidos. Finalmente, en la sección 5 se presentan las conclusiones.

2 FUNDAMENTOS TEÓRICOS

En esta sección se presenta un resumen de las herramientas que utiliza el algoritmo de seguimiento: la representación B-Spline para curvas, el ajuste de un conjunto discreto de puntos por una curva B-Spline y el espacio de formas. Para más detalles ver [2, 23].

2.1 Curvas B-Splines

Dado los puntos de control $\mathbf{Q}_0, \ldots, \mathbf{Q}_{N_B-1}$ donde $\mathbf{Q}_n = (x_n, y_n)^T \in \Re^2$ con $0 \leq n \leq N_B - 1$ y un conjunto de L nodos $\{s_0 < s_1 < \ldots < s_L\} \subset \Re$, una curva B-Spline de orden d se define como la suma ponderada de N_B funciones polinomiales $\mathbf{B}_{n,d}(s)$ de grado $d-1$ dentro del intervalo $[s_i, s_{i+1}]$, $0 \leq i \leq L-1$. La curva B-Spline se construye como $\mathbf{r}(s) = (\mathbf{x}(s), \mathbf{y}(s))^T$ con $0 \leq s \leq L-1$

$$\mathbf{r}(s) = \sum_{n=0}^{N_B-1} \mathbf{B}_{n,d}(s)\mathbf{Q}_n \tag{1}$$

donde $\mathbf{x}(s) = \mathbf{B}(s)^T\mathbf{Q}^x, \mathbf{y}(s) = \mathbf{B}(s)^T\mathbf{Q}^y$ y los vectores de peso $\mathbf{Q}^x$ y $\mathbf{Q}^y$ son las primeras y segundas componentes de $\mathbf{Q}_n$, respectivamente.

El vector de funciones base $\mathbf{B}(s)$ se define como $\mathbf{B} = (\mathbf{B}_{0,d}(s), \ldots, \mathbf{B}_{N_B-1,d}(s))^T$. Si se denota

$$U(s) = \begin{pmatrix} \mathbf{B}(s)^T & 0 \\ 0 & \mathbf{B}(s)^T \end{pmatrix} \tag{2}$$

entonces $\mathbf{r}(s)$ puede escribirse como $\mathbf{r}(s) = U(s)\mathbf{Q}$.

Dado los puntos $D_1, \ldots, D_N$ del plano de la imagen, donde $\mathbf{D}_i = (x_i, y_i)^T, i = 1, \ldots, N$, se desea encontrar la curva B-Spline definida por los puntos de control $\mathbf{Q}_0, \ldots, \mathbf{Q}_{N_B-1}$ que mejor los ajusta. Entonces, se debe satisfacer

$$x_i = \mathbf{B}(t_i)^T\mathbf{Q}^x, y_i = \mathbf{B}(t_i)^T\mathbf{Q}^y \tag{3}$$

para ciertos valores t_i, donde $i = 1, \ldots, N$ y $N_B \leq N$.

Ahora el problema consiste en hallar los puntos de control de manera que la curva B-Spline pase por todos los puntos $\mathrm{D} = \mathrm{K} \begin{pmatrix} \mathbf{Q}^x & \mathbf{Q}^y \end{pmatrix}$, $\mathrm{K} \in \Re^{N \times N_B}$, donde

$$\mathrm{D} = \begin{pmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_N & y_N \end{pmatrix}, \mathrm{K} = \begin{pmatrix} \mathbf{B}(t_1)^T \\ \vdots \\ \mathbf{B}(t_N)^T \end{pmatrix} \tag{4}$$

Definiendo

$$\begin{aligned} t_1 &= 0 \\ t_i &= N_B \frac{\sum_{j=1}^{i} \|D_j - D_{j-1}\|}{\sum_{j=1}^{N} \|D_j - D_{j-1}\|}, 1 < i \leq N \end{aligned} \tag{5}$$

la solución es

$$\begin{pmatrix} \mathbf{Q}^x & \mathbf{Q}^y \end{pmatrix} = \begin{cases} \mathrm{K}^{-1}\mathrm{D} & \textbf{si } N_B = N \\ \mathrm{K}^{+}\mathrm{D} & \textbf{si } N_B < N \end{cases} \tag{6}$$

Si K es cuadrada entonces es inversible. Sin embargo, si $N_B < N$ la matriz no es inversible y por lo tanto se debe calcular la pseudo-inversa.

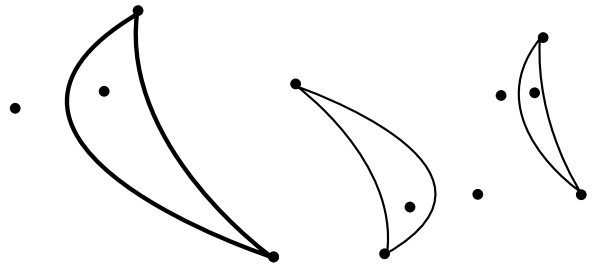


Figura 1: Tranformaciones afines aplicadas a los puntos de control de una curva B-Spline.

2.2 Espacio de Formas

Sea $\mathcal{S} = \mathcal{L}(W, \mathbf{Q}_0)$ el conjunto de las transformaciones y sean $\mathbf{X} \in \Re^{N_\mathbf{X}}$ y $\mathbf{Q} \in \Re^{N_\mathbf{Q}}$ un vector que representa los puntos de control que genera una curva B-Spline. Una transformación $f : \Re^{N_\mathbf{X}} \rightarrow \Re^{N_\mathbf{Q}}$, $f \in \mathcal{S}$ asigna a cada vector $\mathbf{X}$ un vector $\mathbf{Q}$ tal que $f(\mathbf{X}) = \mathbf{Q}$, con $\mathbf{Q} = W\mathbf{X} + \mathbf{Q}_0$ donde $\mathbf{Q}_0$ es la curva patrón o *template* contra la cual se miden las variaciones de las formas y $W \in \Re^{N_\mathbf{Q}} \times \Re^{N_\mathbf{X}}$ es una matriz que corresponde a las transformaciones permitidas de $\mathbf{Q}_0$. Los movimientos permitidos son la traslación, rotación y escalado.

Típicamente ocurre que $N_\mathbf{X} \ll N_\mathbf{Q}$, de manera que se restringe el desplazamiento de los puntos de control a un espacio con menor dimensión. Así, se intenta preservar la forma de la curva, como se ilustra en la Figura 1. En este gráfico se observa el resultado de aplicar las transformaciones permitidas en el espacio de formas a una figura generada por curvas B-Spline (imagen de la izquierda). Al restringir la transformación de la curva a movimientos permitidos en el espacio de formas, se preserva la forma de la curva. Por el contrario, la manipulación arbitraria de los puntos de control no mantiene la forma.

2.2.1 Espacio de las Similitudes Euclideano

Sea $\mathbf{r}_0(s) = U(s)\mathbf{Q}_0$ la curva patrón generada por el vector $\mathbf{Q}_0$. La matriz W que genera el espacio de similitudes euclideanas se define como

$$W = \begin{pmatrix} 1 & 0 & \mathbf{Q}_0^x & -\mathbf{Q}_0^y \\ 0 & 1 & \mathbf{Q}_0^y & \mathbf{Q}_0^x \end{pmatrix} \tag{7}$$

donde la primera y la segunda columna actúan sobre las traslaciones horizontal y vertical, respectivamente. La tercera y cuarta determinan la rotación y la escala a partir del vector $\mathbf{Q}_0$. Es decir, cada columna genera uno de los cuatro movimientos permitidos en este espacio.

Por ejemplo, tomando $\mathbf{X} = (1, 0, 0, 0)^T$ se genera una traslación horizontal sobre la curva $\mathbf{Q}_0$

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & \mathbf{Q}_0^x & -\mathbf{Q}_0^y \\ 0 & 1 & \mathbf{Q}_0^y & \mathbf{Q}_0^x \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{Q}_0^x \\ \mathbf{Q}_0^y \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_0^x + 1 \\ \mathbf{Q}_0^y \end{pmatrix} \tag{8}$$

De la misma manera, $\mathbf{X} = (0, y, 0, 0)^T$, $\mathbf{X} = (0, 0, f_1(\theta), f_2(\theta))^T$ y $\mathbf{X} = (0, 0, z, 0)^T$ determinan una traslación vertical, rotación y escalado isotrópico, respectivamente.

2.2.2 Espacio de Formas Afín Planar

Dadas una curva $\mathbf{r}_0(s)$, una matriz de rotación y escalado $M \in \Re^{2\times 2}$ y un vector de traslación $u = (u_1, u_2)^T$, una transformación afín de $\mathbf{r}_0(s)$ está dada por $\mathbf{r}(s) = u + M\mathbf{r}_0(s)$. El vector u y las coordenadas de la matriz M representan los seis grados de libertad del espacio afín. La matriz de formas W se define como

$$W = \begin{pmatrix} 1 & 0 & \mathbf{Q}_0^x & 0 & 0 & \mathbf{Q}_0^y \\ 0 & 1 & 0 & \mathbf{Q}_0^y & \mathbf{Q}_0^x & 0 \end{pmatrix} \tag{9}$$

Las primeras dos columnas de W representan las traslaciones horizontal y vertical, respectivamente, mientras que las restantes determinan los movimientos de rotación y escalado vertical, horizontal y diagonal. Como $\mathbf{Q} = W\mathbf{X} + \mathbf{Q}_0$, los elementos de $\mathbf{X}$ actúan como pesos en las columnas de W y están dados por $\mathbf{X} = (u_1, u_2, M_{11} - 1, M_{22} - 1, M_{21}, M_{12})^T$. Para más detalles sobre el espacio de formas ver [2].

3 DETECCIÓN DEL CONTORNO DE UN OBJETO

El algoritmo de seguimiento comienza definiendo un área inicial de búsqueda determinada por una curva B-Spline. Luego se plantea una serie de segmentos de recta normales a esta curva y se hallan los puntos de borde del objeto sobre estos segmentos. Finalmente se interpolan estos puntos por una curva B-Spline. A partir de esta estimación inicial, se sigue al objeto utilizando como curva inicial en una imagen, la curva encontrada en el cuadro anterior. Por esa razón es muy importante que el ajuste en el primer cuadro sea muy preciso. El método trabaja restringiendo la transformación de una curva B-Spline a movimientos permitidos dentro del espacio de formas. De esta manera, se preserva la forma de la curva y es un gran ahorro en costo computacional, pero no es posible utilizarlo cuando los movimientos del objeto de interés no son planares. La representación B-Spline tiene la ventaja de que las transformaciones se aplican a los puntos de control en lugar de hacerlo en toda la imagen, lo cual sería excesivamente costoso. El problema ocurre cuando los segmentos de rectas normales a la curva inicial se entrecruzan, provocando que la curva semilla esté muy lejos de la solución óptima. En esos casos el método falla.

3.1 Estimación del Contorno de un Objeto

Para hallar los puntos de borde, se define una curva B-Spline inicial de búsqueda y un conjunto de segmentos normales a ésta. Luego se hallan los puntos de borde que intersecan a estas rectas, como se ve en la Figura 2. Si un punto pertenece al borde del objeto entonces una muestra tomada en una vecindad de ese punto exhibe una discontinuidad en el valor de los niveles de gris y por lo tanto, es un punto de transición. Luego, se encuentra el punto de borde sobre la recta convolucionando los niveles de gris tomados de la imagen con el operador $[-2, -1, 0, 1, 2]$.

Una mala elección de normales puede implicar que los puntos de borde queden invertidos como consecuencia del cruce de normales. Esto introduce errores en la parametrización de la curva, como se ilustra en la Figura 3. Para seleccionar normales no conflictivas, se puede intentar reordenar los puntos de borde. Sea $\overline{\mathbf{R}} = \{\overline{\mathbf{r}}(s_0), \ldots, \overline{\mathbf{r}}(s_N)\}$ un conjunto de puntos de muestra sobre la curva de ajuste. Se aplica el filtro de detección de bordes para encontrar el conjunto de puntos de borde $\mathbf{R}_f = \{\mathbf{r}_f(s_0), \ldots, \mathbf{r}_f(s_N)\}$ sobre la curva característica. Se desea encontrar una función biyectiva $g : \overline{\mathbf{R}} \rightarrow \mathbf{R}_f$, definida por $g(\overline{\mathbf{r}}(s_i)) = \mathbf{r}_f(s_j), 1 \leq i, j \leq N$ tal que no haya cruce de normales, como se muestra en la Figura 4(b). Sin embargo, no está garantizado de que exista g y además, este mecanismo es computacionalmente muy costoso.

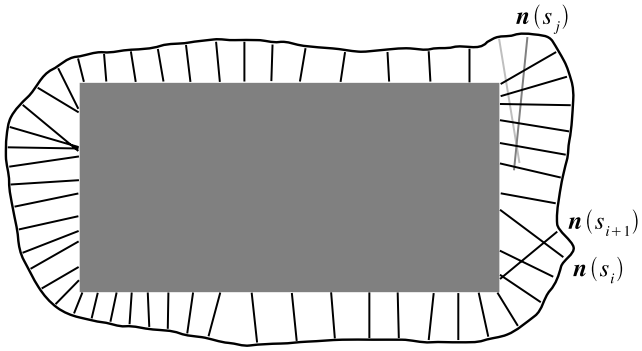
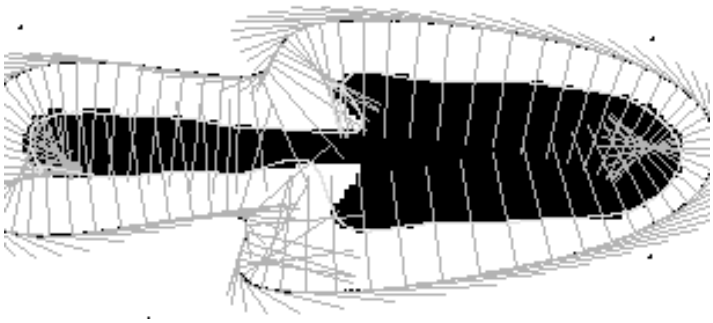
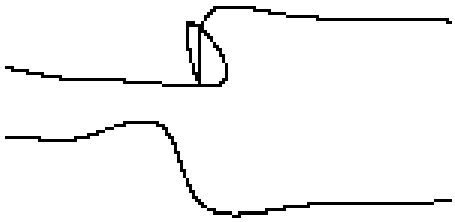


Figura 2: Detección de bordes. A partir de una estimación inicial del contorno del objeto, se trazan rectas normales para encontrar los puntos de borde. $\mathbf{n}(s_i)$ es la normal en el punto s_i.



(a) Generación de rectas normales a la curva inicial.



(b) La curva resultante no ajusta el contorno del objeto como consecuencia del cruce de rectas normales.

Figura 3: El cruce de rectas normales provoca errores en el ajuste del contorno del objeto.

Otra forma de evitar el problema del cruce de normales consiste en analizar secuencialmente el ángulo formado por tres puntos de borde consecutivos, $\mathbf{r}_f(s_i)$, $\mathbf{r}_f(s_{i+1})$ y $\mathbf{r}_f(s_{i+2})$. Se descarta $\mathbf{n}(s_{i+1})$ y se elimina el primer cruce si $ang(\mathbf{r}_f(s_i), \mathbf{r}_f(s_{i+1}), \mathbf{r}_f(s_{i+2})) < 45°$. De esta manera, este mecanismo permite ignorar las normales conflictivas. La Figura 4(c) ilustra esta alternativa.

En este trabajo se propone el método de las *Bounding Boxes* (BB) para solucionar este problema.

Finalmente, una vez hallados los puntos de borde correctos, se construye una curva B-Spline que los ajusta, como se explica en la Sección 2.1.

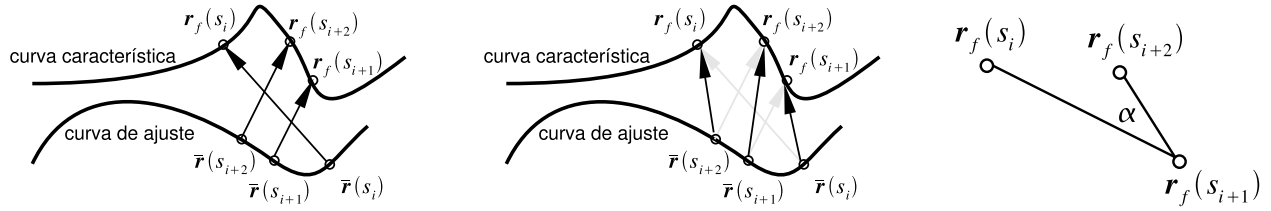
3.2 *Bounding Boxes*

La propuesta consiste en recorrer secuencialmente las normales de a pares. Para cada segmento normal se construye una BB y se realizan cuatro comparaciones. Sea $\mathbf{C}_0^f$ la BB generada por los puntos $\mathbf{C}_0$, $\mathbf{C}_f$ de la siguiente manera

$$\mathbf{C}_0^f = \{P \in I \mid (\text{mín}(\mathbf{C}_0^x, \mathbf{C}_f^x) \leq P^x \leq \text{máx}(\mathbf{C}_0^x, \mathbf{C}_f^x)) \wedge (\text{mín}(\mathbf{C}_0^y, \mathbf{C}_f^y) \leq P^y \leq \text{máx}(\mathbf{C}_0^y, \mathbf{C}_f^y))\} \tag{10}$$

donde I es la imagen. La Figura 5(a) muestra una BB determinada por $\mathbf{C}_0$, $\mathbf{C}_f$.

Sea $\overline{\mathbf{r}}(s)$ la estimación inicial de la forma del objeto en la imagen, $\mathbf{n}(s_i)$ y $\mathbf{n}(s_{i+1})$ dos normales de $\overline{\mathbf{r}}(s)$ muestreadas consecutivamente y P el punto de intersección entre ambas. Para cada i se construye

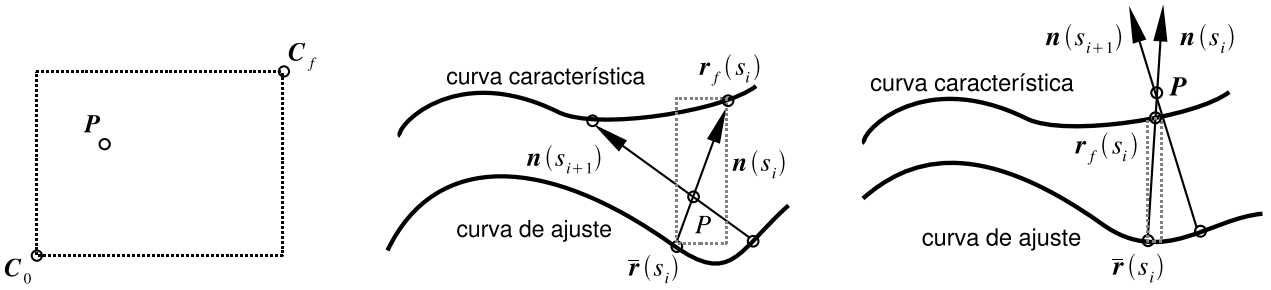


(a) El cruce de normales provoca que puntos de borde queden invertidos.

(b) Función de permutación g tal que $g(\overline{\mathbf{r}}(s_i)) = \mathbf{r}_f(s_{i+1}), g(\overline{\mathbf{r}}(s_{i+1})) = \mathbf{r}_f(s_{i+2}), g(\overline{\mathbf{r}}(s_{i+2})) = \mathbf{r}_f(s_i)$

(c) Ángulo formado por tres puntos de borde consecutivos.

Figura 4: Posibles soluciones al problema del cruce de normales.



(a) Caja Contenedora. (b) $P \in \mathbf{C}^{\mathbf{r}_{f_i}}_{\overline{\mathbf{r}}_i}$. (c) $P \notin \mathbf{C}^{\mathbf{r}_{f_i}}_{\overline{\mathbf{r}}_i}$.

Figura 5: Cajas contenedoras como solución al problema del cruce de normales.

$\mathbf{C}^{\mathbf{r}_{f_i}}_{\overline{\mathbf{r}}_i}$ definida por $\overline{\mathbf{r}}(s_i)$, $\mathbf{r}_f(s_i)$. Si las normales se intersecan antes de alcanzar el borde del objeto, se cumple que $P \in \mathbf{C}^{\mathbf{r}_{f_i}}_{\overline{\mathbf{r}}_i}$, como muestra la Figura 5(b). Entonces se descartan $\mathbf{n}(s_{i+1})$ y $\mathbf{r}_f(s_{i+1})$. En cambio, se considera $\mathbf{r}_f(s_{i+1})$ cuando $P \notin \mathbf{C}^{\mathbf{r}_{f_i}}_{\overline{\mathbf{r}}_i}$, como se ilustra en Figura 5(c).

Una vez seleccionados los puntos de borde, se construye la curva B-Spline que los ajusta.

El Algoritmo 1 muestra el resumen de este proceso de selección de puntos de borde.

Algoritmo 1 Algoritmo de estimación inicial del contorno de un objeto

1: Determinar una región de interés definida por una curva B-Spline.
2: Determinar una serie de segmentos equiespaciados sobre la curva B-Spline.
3: **for all** segmento $s^{(i)}$ **do**
4: Hallar la posición sobre el segmento $s^{(i)}$ donde se encuentra la máxima discontinuidad entre los valores de $s^{(i)}$ convolucionando con el operador $[-2, -1, 0, 1, 2]$.
5: Calcular el punto P de intersección entre el segmento $s^{(i)}$ y el último segmento aceptado S.
6: Construir la caja contenedora C determinada por S.
7: **if** $P \notin C$ **then**
8: $s^{(i)}$ y el punto encontrado se marcan como aceptados.
9: **end if**
10: **end for**
11: Construir la curva B-Spline que interpola los puntos encontrados.

4 RESULTADOS

En esta sección se presentan los resultados obtenidos al aplicar el algoritmo con la modificación del método de las *Bounding Boxes*. La Figura 6 muestra una secuencia de cuadros de un seguimiento sin evitar el cruce de normales. Se observa que las curvas de ajuste en cada cuadro no son aceptables, razón por la cual el algoritmo de seguimiento falla.

En la Figura 7 se observa que las curvas B-Spline de ajuste en cada cuadro son apropiadas y el algoritmo funciona correctamente obteniendo el contorno del objeto en cada cuadro.

5 CONCLUSIONES

En este trabajo se presenta un nuevo método para evitar errores de parametrización debido al cruce de normales. El método de las *Bounding Boxes* permite obtener una curva B-Spline de ajuste muy preciso. Esto es muy importante para inicializar los algoritmos de seguimiento de un objeto en video.

La solución presentada en este trabajo es estable y eficiente ya que requiere cuatro comparaciones para determinar si la intersección entre dos normales consecutivas está dentro del rectángulo generado por una recta. La aplicación del método propuesto presenta resultados visuales muy satisfactorios con un aceptable costo computacional.

REFERENCIAS

[1] A. Agarwala, A. Hertzmann, D. H. Salesin, and M. S. Steven. Keyframe-based tracking for rotoscoping and animation. *International Conference on Computer Graphics and Interactive Techniques. ACM SIGGRAPH 2004*, 2004.

[2] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.

[3] P. Brigger, J. Hoeg, and M. Unser. B-Spline snakes: A flexible tool for parametric contour detection. *IEEE Trans. on Image Processing*, 9(9):1484–1496, September 2000.

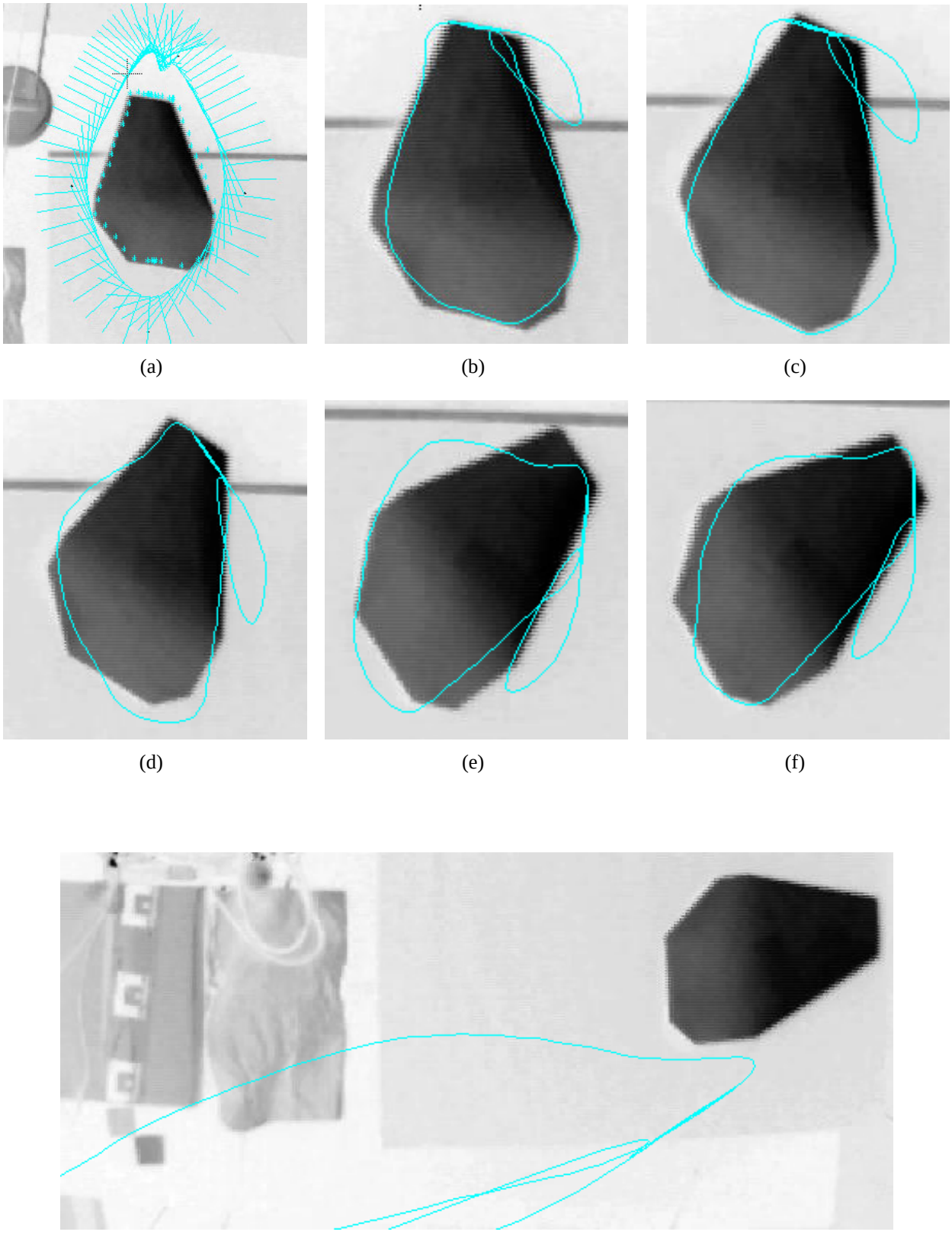
[4] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Proceedings of the 3rd International Conference on Computer Vision*, pages 616–625, 1990.

[5] E. Debreuve, M. Barlaud, G. Aubert, I. Laurette, and J. Darcourt. Space-time segmentation using level set active contours applied to myocardial gated SPECT. *MedImg*, 20(7):643–659, July 2001.

[6] F. Derraz, M. Beladgham, and M. Khelif. Application of Active Contour Models in Medical Image Segmentation. *International Conference on Information Technology: Coding and Computing (ITCC'04)*, 2:675, 2004.

[7] P. Dickson, J. Li, Z. Zhu, A. Hanson, E. Riseman, H. Sabrin, H. Schultz, and G. Whitten. Mosaic Generation for Under Vehicle Inspection. In *Workshop on Applications of Computer Vision*, Orlando, USA, December 2002.

[8] P. F. Felzenszwalb. Representation and Detection of Deformable Shapes. In *CVPR (1)*, pages 102–108, 2003.

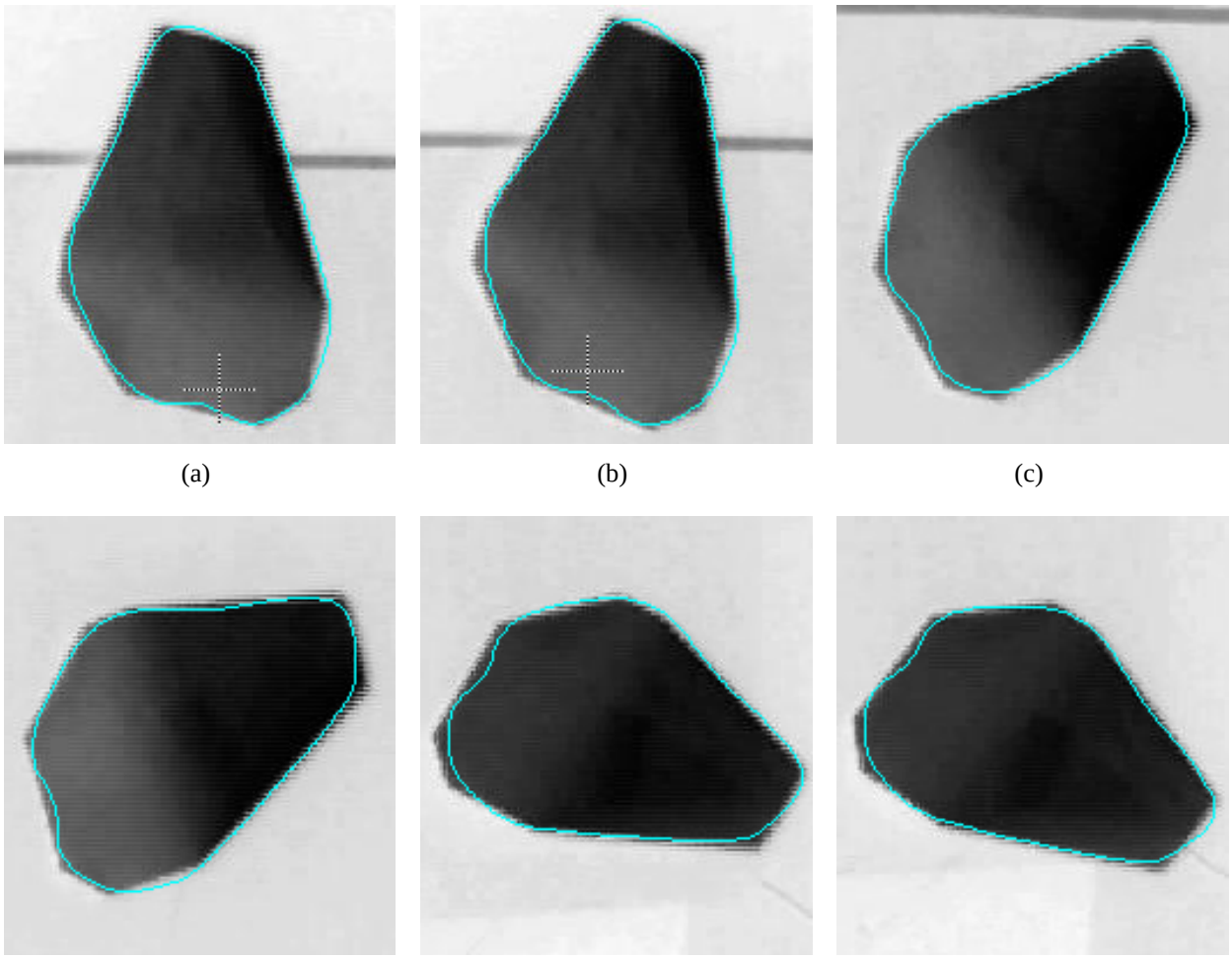


(a) (b) (c)

(d) (e) (f)

(g)

Figura 6: Seguimiento del objeto sin evitar el cruce de normales. El algoritmo es incapaz de seguir al objetos debido a los problemas en el ajuste.



(a) (b) (c)

(d) (e) (f)

Figura 7: Seguimiento del objeto utilizando el método propuesto.

[9] A. Fournier and P. Poulin. A ray tracing accelerator based on a hierarchy of 1D sorted lists. In *Proceedings of Graphics Interface '93*, pages 53–61, Toronto, Ontario, May 1993. Canadian Information Processing Society.

[10] J. Gambini, M. Mejail, J. Jacobo Berllés, and A. Frery. Feature Extraction in Speckled Imagery using Dynamic B-Spline Deformable Contours under the $\mathcal{G}^0$ Models. *International Journal of Remote Sensing*, 27:5037–5059, November 2004.

[11] M. J. Gambini, M. E. Mejail, J. Jacobo Berlles, and A. Frery. Polarimetric SAR Region Boundary Detection using B-Spline Deformable Contours under the $\mathcal{G}$ Model. In *XVII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI) / II Ibero-American Symposium on Computer Graphics (SIAGG)*, pages 764–769, Natal, Brasil, October 2005.

[12] A. S. Glassner. *An Introduction to Ray Tracing*. Academic Press, London, 1989.

[13] F. Huang and J. Su. Deformable Pedal Curves with Application to Face Contour Extraction. In *CVPR (1)*, pages 328–333, 2003.

[14] J. Denzler and H. Niemann. Active Rays: Polar-transformed Active Contours for Real-Time Contour Tracking. *Real-Time Imaging*, 5(3):203–213, 1999.

[15] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Model. *International Journal of Computer Vision*, 1(1):321–333, March 1988.

[16] A. Katz, D. Wassermann, J. Gambini, J. Jacobo, and M. Mejail. Real Time object tracking for Mirosot League. In *FIRA Robot World Congress*, 2003.

[17] D. Kim. B-Spline Representation of Active Contours. In *Fifth International Symposium on Signal Processing and its Applications, ISSPA99*, Brisbane, Australia, 1999.

[18] S. Lefèvre and N. Vincent. Real Time Multiple Object Tracking Based on Active Contours. In *ICIAR (2)*, pages 606–613, 2004.

[19] J. Mahovsky and B. Wyvill. Fast ray-axis aligned bounding box overlap tests with Plücker coordinates. *journal of graphics tools*, 9(1):35–46, 2004.

[20] S. Malassiotis and M. G. Strintzis. Tracking the left ventricle in echocardiographic images by learningheart dynamics. *IEEE Transactions on Medical Imaging*, 18:282–290, March 1999.

[21] G. Medioni and Y. Yasumoto. Corner detection and curve representation using curve B-Splines. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 764–769, 1986.

[22] F. Precioso, M. Barlaud, T. Blu, and M. Unser. Smoothing B-Spline Active Contour for Fast and Robust Image and Video Segmentation. In *Proceedings of the 2003 IEEE International Conference on Image Processing (ICIP'03)*, pages 137–140, Barcelona, Spain, September 2003.

[23] D. F. Rogers and J. A. Adams. *Mathematical Elements for Computer Graphics*. McGraw-Hill, New York, 2nd edition, 1990.

[24] H. Schmidl, N. Walker, and M. Lin. CAB: Fast update of OBB trees for collision detection between articulated bodies. *journal of graphics tools*, 9(2):1–9, 2004.

[25] S. Suri, P. M. Hubbard, and J. F. Hughes. Analyzing bounding boxes for object intersection. *ACM Transactions on Graphics (TOG)*, 18(3):257–277, July 1999.

[26] D. Wassermann, M. Mejail, J. Gambini, and M. E. Buemi. Segmentation with active contours: a comparative study of B-spline and level set techniques. In *SIBGRAPI XVII:17th Brazilian Symposium on Computer Graphics and Image Processing*. IEEE, 2004.

[27] G. Zachmann. Minimal Hierarchical Collision Detection. In *Proc. ACM Symposium on Virtual Reality Software and Technology (VRST)*, pages 121–128, Hong Kong, China, November11–13 2002.

[28] Z. Zhu, G. Xu, E. Riseman, and A. Hanson. Fast Generation of Dynamic and Multi-Resolution 360 Degrees Panorama from Video Sequences. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 400–406, Florence, Italy, 1999.

[29] A. Zomorodian and H. Edelsbrunner. Fast software for box intersections. *International Journal of Computational Geometry and Applications*, 12(1-2):143–172, 2002.

Stream programming Framework for Global Ilumination Techniques Using a GPU

Federico J. Marino
Facultad de Ingeniería, Universidad de Buenos Aires
Av. Paseo Colón 850 (C1063ACV), Capital Federal, República Argentina
fedemarino@gmail.com

and

Horacio Abbate
Facultad de Ingeniería, Universidad de Buenos Aires
Av. Paseo Colón 850 (C1063ACV), Capital Federal, República Argentina
habbate@fi.uba.ar

Abstract

Stream processors are becoming an affordable alternative to implement hardware assisted rendering techniques which were usually relegated to offline usage. We built a stream processing framework based on the Stream Programming Model concepts, selected the Photon Mapping algorithm and an NVIDIA GPU (Graphics Processing Unit) as a test case implementation of a Global Illumination technique. We defined a set of C++ classes to encapsulate the components (kernels and streams) of this new paradigm, using OpenGL and Cg language. Our application combines the Photon Splatting method and the BVH (Bounding Volumes Hierarchy) acceleration structure into a rendering pipeline relying almost entirely on the GPU. Finally, we evaluated its performance using a Cornell Box model.

Keywords: Photon Mapping, GPU, Stream Programming, Photon Splatting, Cg, OpenGL, BVH, Real Time Rendering, Global Illumination.

Resumen

Los procesadores de streams están comenzando a ser una alternativa accesible para implementar técnicas de rendering asistidas por hardware que habitualmente estaban relegadas al uso offline. Nosotros elaboramos un marco de trabajo para procesamiento de streams basado en los conceptos del modelo de Stream Programming, seleccionamos el algoritmo de Photon Mapping y una GPU (Graphics Processing Unit) Nvidia para una implementación de un caso de prueba. Definimos un conjunto de clases en C++ para encapsular los componentes (kernels y streams) de este nuevo paradigma, usando OpenGL y el lenguaje Cg. Nuestra aplicación combina el método de Photon Mapping y una estructura de aceleración BVH (Bounding Volumes Hierarchy) en un pipeline de renderizado basado casi completamente en la GPU. Finalmente, evaluamos su desempeño usando un modelo de caja de Cornell.

Palabras claves: Photon Mapping, GPU, Stream Programming, Photon Splatting, Cg, OpenGL, BVH, Rendering en Tiempo Real, Iluminación Global.

1. BACKGROUND

Modern programmable graphics processors (GPUs) integrate multiple processing pipelines working in parallel, with a high-speed dedicated memory (texture memory) that can achieve a combined performance of many gigaflops. Their processing power grows at a higher rate that traditional CPUs, motivating a change of paradigm in the programming techniques used for compute-intensive and real-time applications.
GPUs consist of vertex and fragment processing units capable of handling high precision arithmetic. Vertex processors handle the transformations and projections of geometry vertices of the 3D model which are then provided to the rasterizer, which converts vector data (triangles) into bitmap data (fragments). The fragment processors do the shading of the triangles, applying an illumination model specified by a fragment program (written in a C-like language, called Cg [11]). They read data stored in textures and usually produce pixel colors as results.
Global illumination algorithms aim to simulate a physically correct illumination of a 3d scene through a numerical approximation to the solution of the rendering equation [1]. Some years ago, Photon Mapping appeared as a unified technique able to simulate most of the effects of light-surface interactions in an efficient and elegant way, covering a wider range of situations than previous approaches such as, Radiosity or Path-tracing. Figure 1 shows a sample scene with and without global illumination.
In order to adapt an algorithm like Photon Mapping to this type of hardware, we have to use the approach known as GPGPU (general purpose computing on GPU). The idea is to consider the inputs and outputs of the fragment processor as general data instead of image-specific. Data structures have to be mapped into textures and code supported by fragment or vertex programs written Cg. Although the programmability on recent GPU generations has improved, flow control and branching instructions have performance penalties. The lack of scatter operations (random access writes) is another limiting factor, especially for general algorithms like sort.

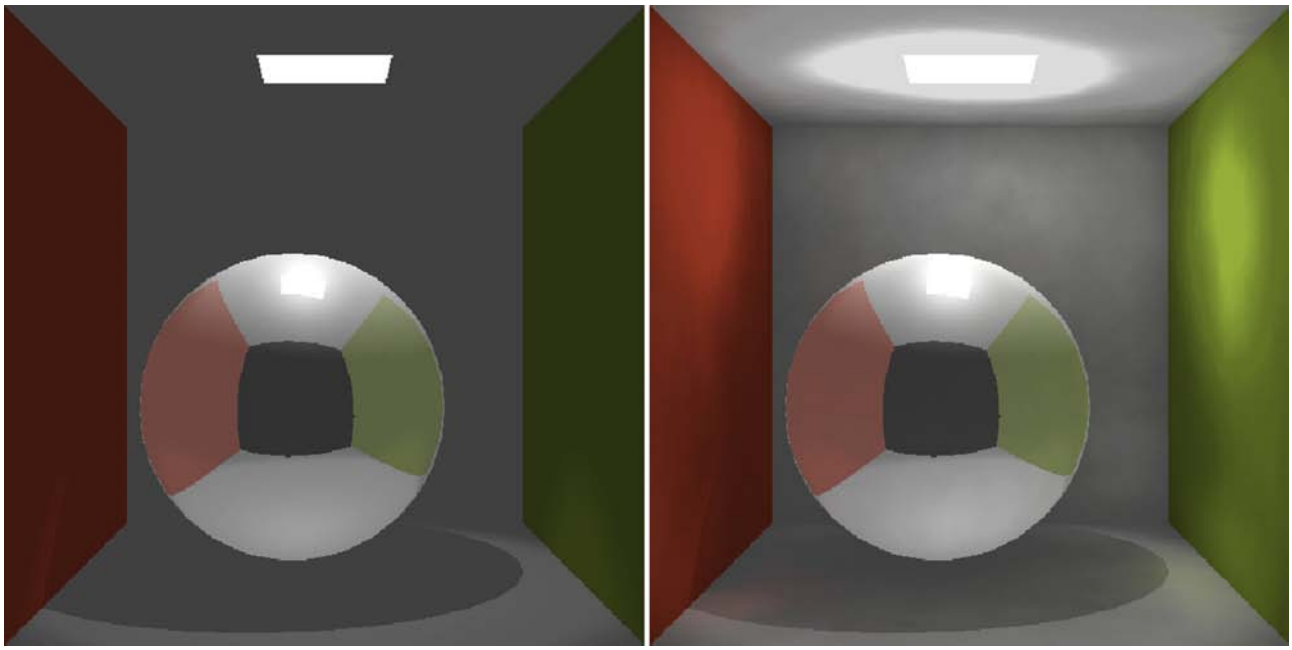


Figure 1: Sample Scene, without global illumination (left), with global illumination (right)

Photon mapping was developed by Henrik Wann Jensen [2] and can be classified as a density (of particles per surface area unit) estimation two-phase algorithm. The term photon has its origin in the physical term, but in this case it only represents a particle carrying a discrete amount of luminance energy on a specific direction. Other term used is radiance, which defines the radiant energy flux per area unit in a certain direction. In addition, the term irradiance refers to the integral of the radiance over a hemisphere (radiant flux emitted per unit area in all directions).
The first phase of Photon Mapping consists on distributing light particles (photons) from the source in random directions, through the 3d scene, until the maximum number of interactions is reached. On each photon-surface interaction, the Russian roulette technique is used to decide probabilistically whether photons should be reflected, refracted or absorbed.
The traced photons are stored in a data structure called photon map, including data such as hit coordinates on the surface, incident direction and photon's energy.
The second phase uses a standard ray tracing algorithm to trace primary rays from the viewer's eye into the scene through the pixels of the view plane (one ray per pixel). For each pixel, the amount of indirect illumination is calculated using the radiance estimate, which gathers the nearest neighbor photons of the surface hit point and sums its contribution to the radiance, applying the respective BRDF (Bidirectional Reflectance Distribution Function) [12] of the surface material.
One remarkable feature of Photon Mapping is that the illumination representation (photon map) is decoupled from geometry, so that illumination detail is not tied to 3D geometry level of detail.
The results of the method are biased, meaning that because they are based on a number of pre-computed particles, they will never converge to the real value of the integral it is attempting to solve; even though, the method is consistent.
There are several optimization strategies, such as irradiance caching proposed by Ward [3], who introduced the idea of re-using irradiance values in certain points of lambertian surfaces via interpolation.

2. ILLUMINATION TECHNIQUES ON GPUS

Thrane and Simonsen [7] proposed the BVH acceleration structure to speedup the ray tracing process. The idea is to organize the geometry in a binary tree forming a bounding volumes hierarchy. Therefore, many ray-triangle tests can be avoided by first testing the intersection with the ray and the box bounding a group of triangles. This structure has proved to be more efficient and easier to implement on a GPU compared to KD-trees or Uniform Grids.
Lavignotte and Paulin [6] presented a new image based method for computing an efficient global illumination solution using graphics hardware. To calculate the irradiance they propose the Photon Splatting technique, which distributes each photon's contribution over the pixels instead of gathering radiance from the neighborhood (which requires some complex search structure difficult to implement on a GPU). At the moment they presented their work, hardware limitations (8-bits per channel precision buffers) required complex workarounds which now have been surpassed.
Moelhave [4] implemented a ray tracing engine on the GPU, using a uniform grid acceleration structure. It was based on Pixel buffers extension, OpenGL and C++. That implementation was only capable of rendering direct illumination of diffuse surfaces without reflections, refractions or shadows support.

3. STREAM PROGRAMMING FRAMEWORK

Our approach is based on the concepts exposed by Purcell [5], the Stream Programming Model, the MRT (Multiple render targets) feature and the FBO (Frame Buffer Object) OpenGL extension. The MRT feature allowed us to output up to four 32-bit floats from a fragment program, reducing the

need of multiple runs of the same kernel to evaluate each output separately. The FBO extension is a replacement for Pixel Buffers (PBuffers) that allows an efficient and flexible use of the render-to-texture mode in OpenGL.

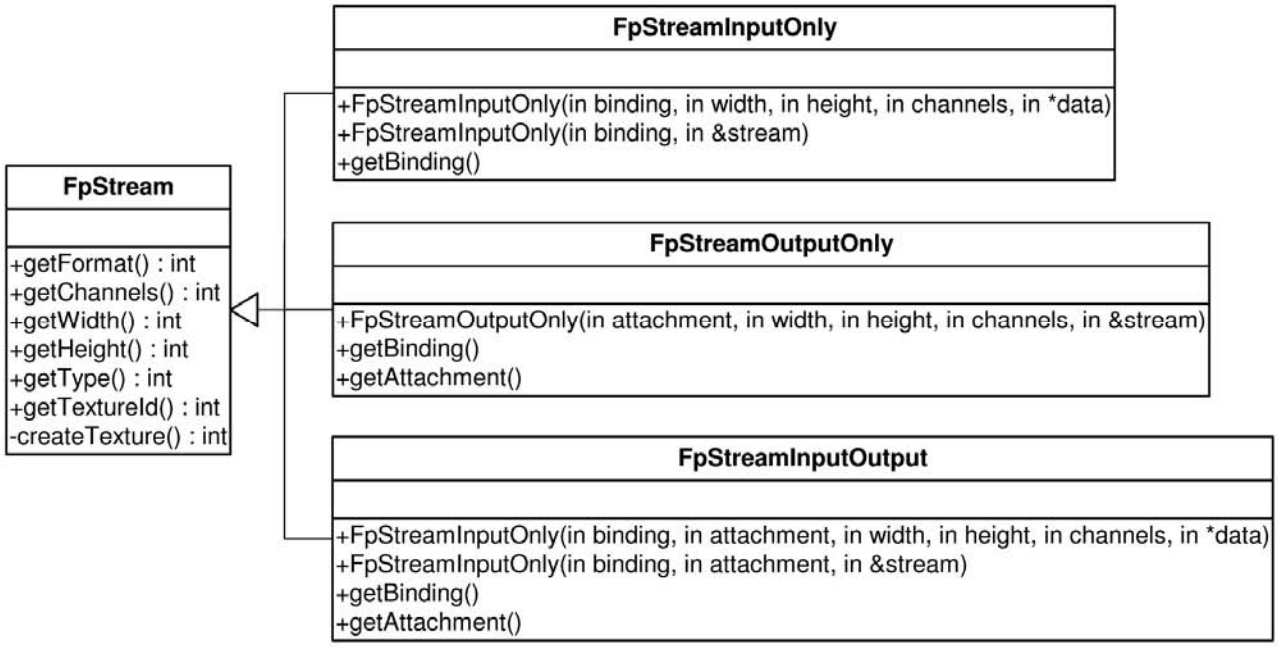


Figure 2. The FpStream class

One of the components of the Stream Programming Model is the kernel, which is basically a function call that performs a considerable amount of computations on a set of records known as streams. A kernel takes streams as inputs and produces streams as outputs.
We defined a set of C++ classes (FpStream and FpKernel) to encapsulate the render-to-texture mechanisms used to compute the results. The FpStream class (see figure 2) holds a 2D texture with up to 4 channels (RGBA) of 32 bits floating point precision, where application data structures fit in.

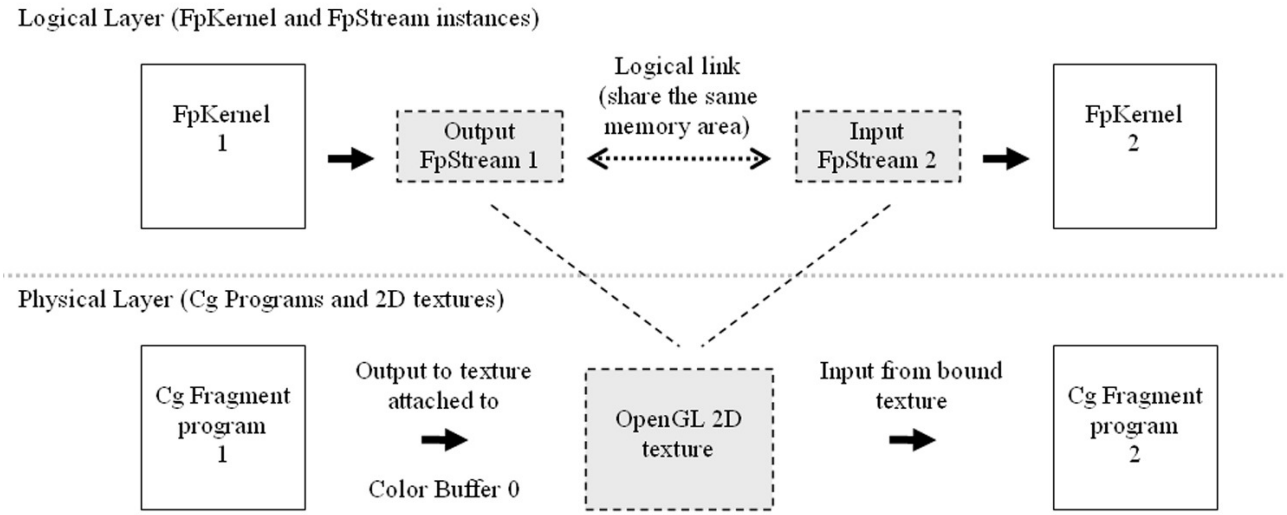


Figure 3. Framework layers.

Three child subclasses represent the modes in which the streams can be linked to a kernel: output, input or input-output modes. The constructor method of an input stream can receive an output stream as a parameter, providing a way to link one kernel output to other kernel's input or vice versa. The link is done logically, meaning that the actual data is represented only by one texture id, shared by those two streams that are used for writing and reading respectively (see figure 3).
It is also possible to have multiple input streams sharing the same physical texture, in cases where the same data serves as input for multiple kernels, in different contexts.
The FpKernel class (see figure 3) is associated to Cg fragment program. This class can represent both single-pass and multi-pass kernels. The first step is to bind the uniform parameters (global static parameters for a Cg program) to it. The second step is to bind the streams providing a "binding" name for the inputs which is the name used by the stream within the CG program, and specifying an "attachment" number for the outputs (this number corresponds to the output color buffer of the Cg program, which should be between 0 and 3).
Finally, the computation is carried out by the "run" (see figure 4 left) method which binds a FBO (Frame Buffer Object) instance, sets an orthographic projection frustum and then draws a rectangle (GL_QUAD) parallel to the view plane which covers it from the lower-left to the upper-right corner (see figure 4 right). The rasterizer converts the rectangle geometry into fragments, on which the Cg program runs and then the outputs are stored into the off-screen buffer (output streams).
Since GPUs process stream elements (fragments) independently, it's impossible to share the fragment processing results among them. This is due to the fact that fragment processors work in parallel running the same program over different parts of the stream in no particular order.

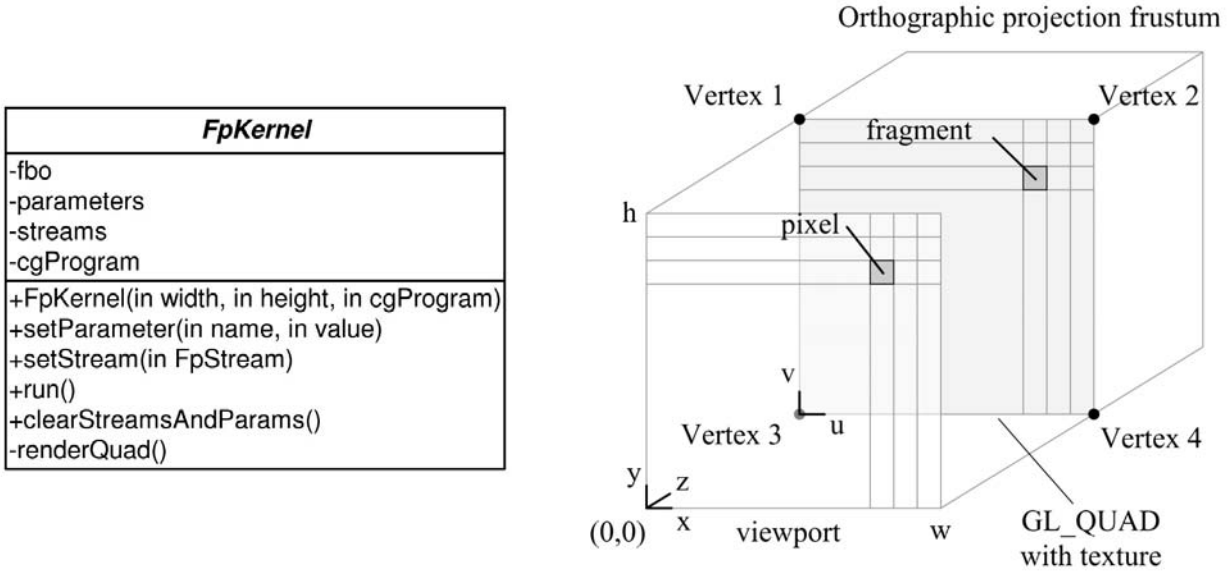


Figure 4. FpKernel class (left), Kernel computation (right).

When running multi-pass algorithms (which require many iterations of the same kernel), some elements of the streams could reach the end state of the process earlier than others. But current GPUs do not allow you to selectively disable the execution of the kernel on certain stream elements. Instead, you can only prevent the output value for a certain fragment to be written to the buffer, by using the discard instruction. To detect the condition when all stream elements have reached the end state, the OCCLUSION_QUERY extension is used. It returns the number of fragments that were updated in the output buffer or active in a certain pass. A loop over the kernel "run" method is done until this number reaches zero.

4. THE PHOTON MAPPING IMPLEMENTATION

Surface materials are modeled by the sum of two types of BRDF functions, ideal diffuse and specular reflections.
The light source is modeled as a directional spotlight, defined by an origin, direction, minimum and maximum angles which define two cones between which the intensity varies from maximum light power to total darkness.
In the photons tracing phase, the total power of the light is evenly spread on the generated photons. The color of the photons is filtered by the surface color they hit (color bleeding effect). Photons can be absorbed, diffusely reflected or specularly reflected when they hit a surface according to specularK or diffuseK coefficients. In figure 5 (left), photon A is absorbed after 2 reflections, photon B bounces into space and gets discarded, finally photon C is absorbed after 1 specular reflection.
During the rendering phase (see figure 5), direct illumination component is calculated using the Phong model [8] including the calculation of a shadow factor by tracing one shadow ray from the primary ray hit point to the light source. If the shadow ray hits a surface before light source, the shadow factor is zero.Otherwise, it is set to 1.0. To compute the indirect illumination, we estimate the irradiance at a surface point by splatting the photons (see point 4.2) and compute mirror-like reflections by tracing a secondary ray and adding its color according to the reflectivity coefficient of the material. The irradiance value is independent of the outgoing direction we are analyzing, so it is stored in a scalar variable which accumulates photons contributions.

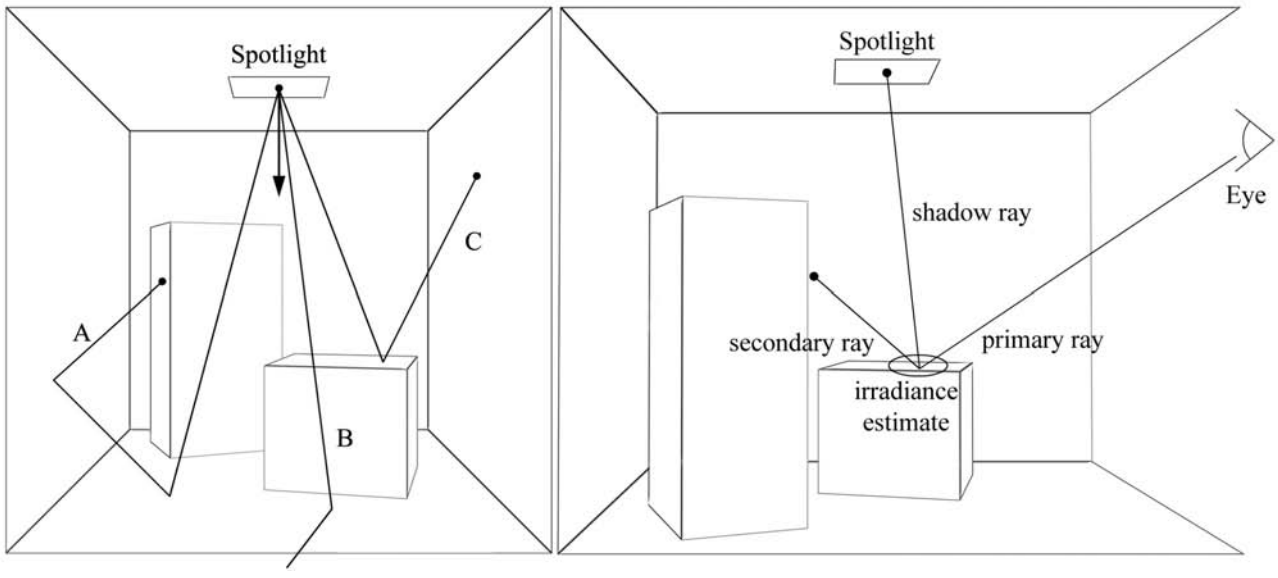


Figure 5. Photon tracing (left) & Rendering phase (right).

4.1. Photons and Rays Tracer

The process of tracing rays or photons is almost similar. The differences lay on the results of the interaction with the surface they hit, so basically, the algorithm and the acceleration structure can be shared for both purposes.
When using BVH, the geometry is organized in a binary tree, by recursively subdividing the space in two parts along one axis of the three axes (X, Y and Z), until each individual triangle is enclosed in the bounding box.

Actually, the tracing process does not require the tree itself, but an array specifying a sequence of nodes to traverse it, starting at the root node. Following that sequence a ray-triangle intersection should be found faster than using a uniform grid depending on the distribution of the triangles in the space.
The construction of the tree is done entirely on the CPU as a pre-process. In the case of a static scene (geometry does not change over time) no tree reconstruction is required.

4.2. The Splatting Process

This technique was presented by Stürzlinger and Bastos [9]. A splat has the shape of a disc and represents the photon contribution to indirect illumination in the surroundings of the hit point on a surface. For each photon stored, a splat should be placed centered on the hit point and oriented in the 3D space. After tracing the photons on the GPU, the output streams containing photons data are transferred and converted to arrays in the CPU memory, because geometry (triangles to represent the splats) can't be generated as an output from a fragment program. A CPU process scans the array of photons and draws a triangle holding the splat into the framebuffer using alpha blending mode, so that all the photon contributions are accumulated.
During the shading process of the triangle where the splat is supported, each fragment is tested, analyzing whether it belongs to the same surface that the photon has hit. If not, its contribution is discarded. In figure 6, it is observed that the photon contributes to the illumination of fragment A but not to the fragment B. The decision is based on comparisons between the normal vectors and the surface identifiers.

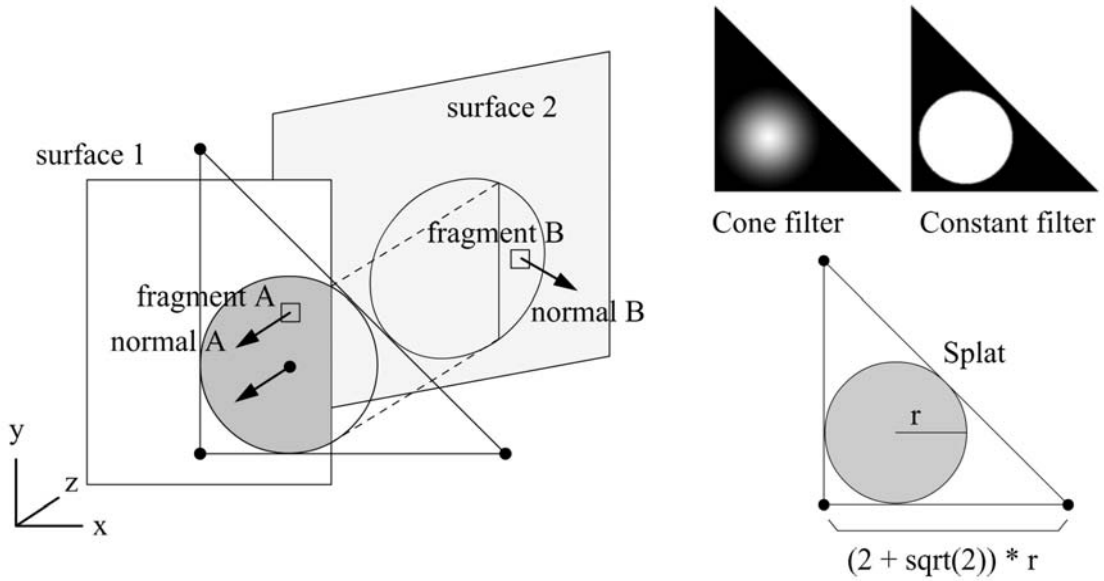


Figure 6. Splatting process (left), splat shape and filter types (right).

The energy provided by each photon to each fragment is weighted by a function (known as filter) that depends on the distance to the center of the splat (see figure 6). Typically, a constant filter, a cone filter (intensity decays linearly until the radius is reached) or gaussian filter is used.
The contribution from all the splats is accumulated in the framebuffer. When a large number of splats are added, a smooth indirect illumination image is obtained (figure 7 shows the splatting process). The resulting texture is the called irradiance image and serves as input of the second phase of the algorithm (the rendering phase).

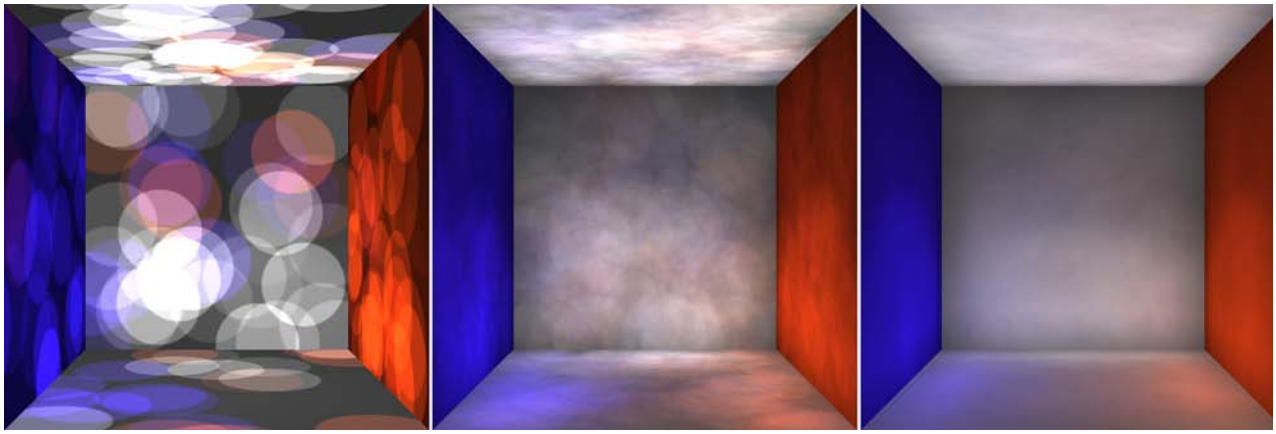


Figure 7. Splatting process with 256 photons (left), 4K photons (center), 64K photons (right)

5. THE PHOTON MAPPING ENGINE

A set of classes (SceneLoader, Tracer, PhotonMapper and RayTracer) represents the chain of processes of the algorithm, where the inputs are the scene geometry, illumination and view parameters, and the output is the synthesized image as a bitmap file (see figure 8). The SceneLoader class provides the geometry and builds the BVH tree, in the form of streams, from a 3DS format file (3DS Studio) and a plain text file that includes global parameters, surface materials, view frustum and light properties.

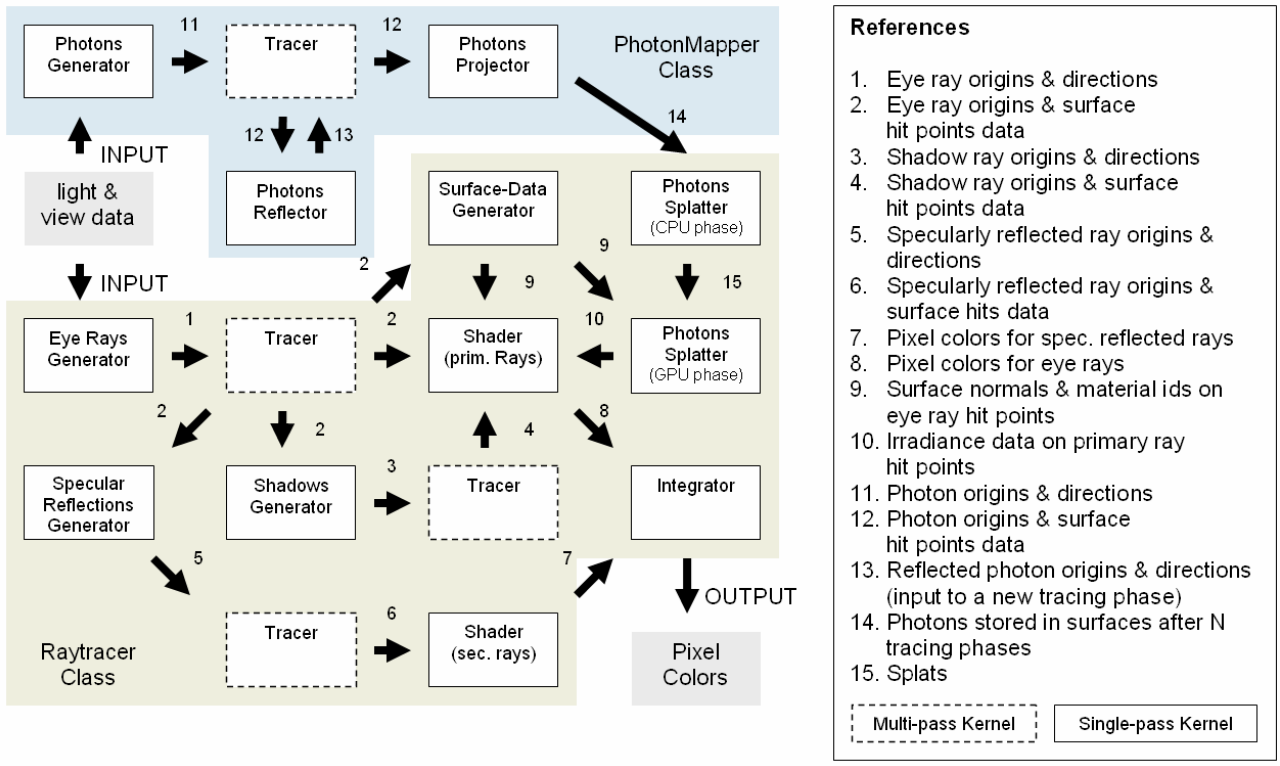


Figure 8. Classes layout.

The Tracer class is responsible for tracing both eye rays and photons through the scene using BVH acceleration structure. It receives the origin and direction vectors as inputs, and outputs the coordinates of the triangle hit points.
The PhotonMapper class uses the services of the tracer class. A kernel named Generator creates the photons, sampling the spotlight emission space (using random polar coordinates). Photons are traced up to a maximum number of bounces. After each tracing step, the Reflector kernel decides whether each photon should be reflected or absorbed based on the respective coefficients of the surface material. If a photon is reflected, its new instance is fed again to the Tracer. Finally, the absorbed photons are projected (Projector Kernel) in world space coordinates and transferred to the CPU memory (Raytracer instance).
The Raytracer class handles the splatting process (to generate the irradiance image) and rendering phase using the services of the Tracer class. It is responsible for combining all the illumination components (ambient, direct, indirect, mirror-like reflections and shadows) to deliver the final image.
Figure 9 shows sample images rendered with our application using 262K photons. The lighting on the walls exhibits low frequency noise due to the still insufficient number of photons.

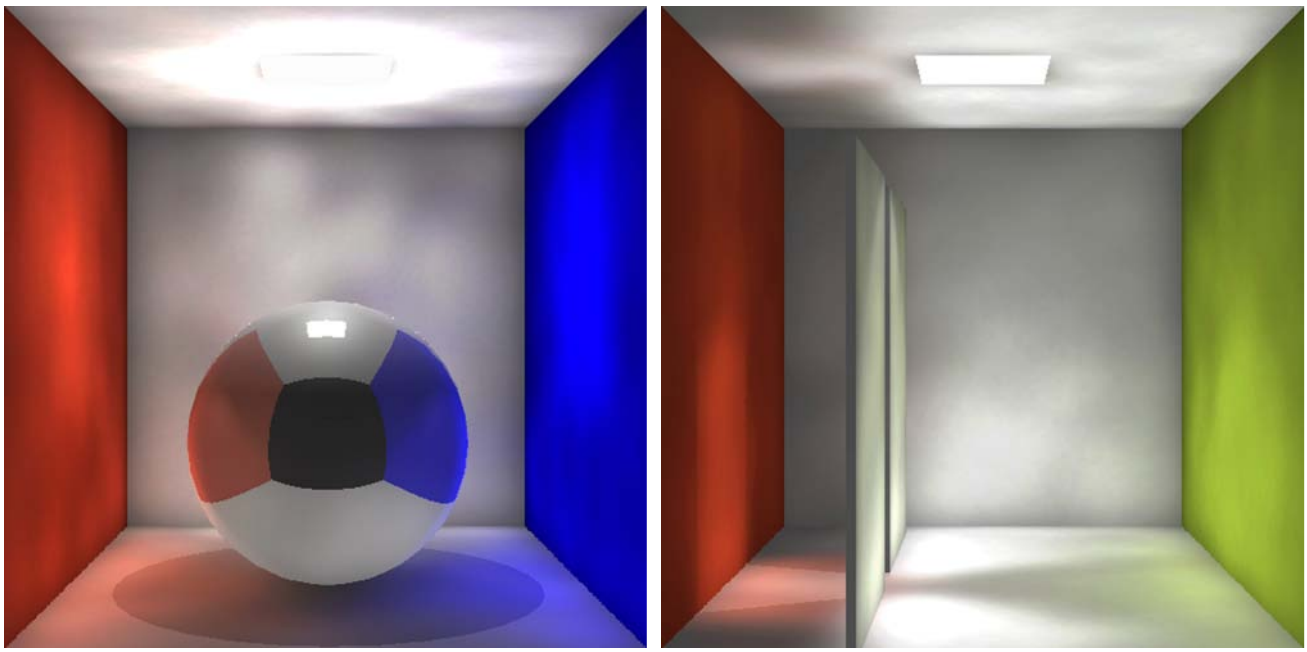


Figure 9. Sample scenes.

6. PERFORMANCE TESTS

A Cornell Box model was used to evaluate our implementation (see figure 10). All the surface materials are non-reflective (no mirror-like reflections) and have a diffuse reflection coefficient of 0.2 and a specular coefficient of 0.1, so, if 100 photons intersect a certain surface , 70% will be absorbed by it, 20 % will be diffusely reflected and the remaining 10% will be specularly reflected. Photons were traced up to a maximum of 3 bounces (4 tracing phases).
It's important to note that not all generated photons produce a splat, because some of them are reflected outside the scene and others are lost after being reflected past the photons bounces limit. Empirically, on this scene, the number of generated splats is between 70% and 75% of the generated photons. The number of splats varies, even on consecutive runs with the same input parameters because it depends on randomly generated numbers.

Figure 10. Scene used in performance tests (left), other sample scene (right).

The hardware and software platform used was Microsoft Windows XP SP2, Visual C++ 7.0, OpenGL 2.0, NVIDIA Cg 1.5 toolkit, an NVIDIA GeForce 6600 GTS 128MB AGP GPU, an AMD Athlon 64 3000+ CPU with 1GB DDR RAM and NVIDIA driver version 91.45. We used the fp40 and vp40 Cg profiles.
The following 27 parameters combinations were evaluated using the test scene:

- Image resolution: 128x128, 256x256 and 512x512 pixels.
- Splat radius: 0.5, 1.0 and 1.5 units (as a reference, the back wall is 10 units wide).
- Generated photons: 16384, 65535 and 262144 (approximately. 75% are splatted).

Globally, it is observed that the implementation is scalable as the rendering time increases sub linearly vs image resolution, splat radius and number of photons (see table 1). It is expected that with future GPU generations, the increased number of processing units will produce a direct performance boost, as more streams will be processed in parallel per clock cycle.
Table 2, shows a second set of tests, using a splat radius of 1.25 units and 512x512 image resolution. The results show that the splatting and tracing phases have an important share in the total computational cost specially when using 262144 generated photons where the quality of the solution seems acceptable (figure 10 left corresponds to that case).

Table 1. Rendering Time

Generated photons	**Total rendering time (seconds) according to image resolution (pixels)**								
	Splat radius = 0.5			**Splat radius = 1.0**			**Splat radius = 1.5**		
	128x 128	**256x 256**	**512x 512**	**128x 128**	**256x 256**	**512x 512**	**128x 128**	**256x 256**	**512x 512**
16384	4,87	5,09	6,17	5.47	5.11	6.78	4.86	5.19	6.65
65535	5,87	6,56	7,37	5.89	6.25	8.22	6.01	6.53	9.48
262144	10,67	11,43	12,33	10.69	11.,22	15.89	10.71	12.34	20.43

Table 2. Splatting and tracing phase incidence.

Generated Photons	Time (seconds)		
	Total	Photons Tracing	Photons Splatting
1024	6.81	2.78	0.21
4096	6.89	2.81	0.27
16384	7.42	2.98	0.62
65535	9.42	3.71	1.67
262144	16.92	7.26	5.51

7. CONCLUSIONS AND FUTURE WORK

As seen in the tests results, the splatting phase consumes much of the rendering time (table 2), so it would be advisable to concentrate future efforts on optimizing it. One way to do it is to study the benefits of using rectangles (GL_QUAD) or points (GL_POINT) instead of triangles (GL_TRIANGLES) for representing the splats, thus, less time would be wasted processing fragments that are outside the splat disc.

Another interesting issue would be to investigate the use of the "geometry instancing" feature, currently only supported under Direct3D. That feature allows us to generate geometry, by clonning and transforming geometry that is already stored in GPU memory (in our case, the splats), instead of creating them one by one by reading the photon's list.

In addition, it may be useful to analyze the possibility of running the process of photon tracing and ray tracing simultaneously using two GPUs connected in the same PC using a SLI (Scalable Link Interface) configuration, as well as taking advantage of the idle periods of the CPU while the GPU is working on relatively long processes, perhaps via multiple threads.

The next step towards building interactive Photon Mapping engine with real time feedback would be to study the way to reuse calculations between consecutive frames. For example, if lighting conditions do not change, photons do not need to be retraced. When lighting conditions change partially, only some of the photons could be retraced using the strategy called Selective Photon Tracing [10].

Finally, new emerging technology such as NVIDIA CUDA (Compute Unified Device Architecture) seems to be a more elegant and powerful base platform to build a stream processing application like the one we have studied. This new architecture has an API specifically designed for general-purpose computation, where code can be written directly in C language. It also allows the use of scatter instructions (that limitation in our implementation forced us to run part of the splatting process on the CPU, generating costly memory transfers through the AGP bus).

REFERENCES

[1] J. T. Kajiya, "The Rendering Equation", Proceedings of ACM SIGGRAPH 86, 1986, pages 143-150.

[2] H. W. Jensen, "Realistic Image Synthesis using Photon Mapping", A K Peters, 2001. ISBN 1568811470.

[3] G. J. Ward, F.M Rubinstein and R. D. Clear, "A ray tracing solution for diffuse interreflection", Computer Graphics (Proc. SIGGRAPH '88) 22(4): 85-92 (August 1988).

[4] T. Moelhave, "Implementing a Ray Tracer on a GPU", http://moelhave.dk/gpu-ray-tracer/, Jan. 2005.

[5] T. J. Purcell, "Ray Tracing on a Stream Processor", PhD thesis, Stanford University, 2004.

[6] F. Lavignotte, and M. Paulin, “Scalable Photon Splatting for Global illumination”, Graphite 2003 (International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia), Melbourne, Australia, ACM SIGGRAPH, pp. 1-11.
[7] N. Thrane, L. O. Simonsen, “A Comparison of Acceleration Structures for GPU Assisted Ray Tracing”, Master’s thesis, University of Aarhus, 2005.
[8] Phong, B.T.: Illumination for computer generated pictures. Comunnications of ACM 18, 6, ISSN 0001-0782, (Junio 1975),pp 311 – 317.
[9] W. Stürzlinger, R Bastos, “Interactive Rendering of Globally Illuminated Glossy Scenes”, Eurographics Rendering Workshop ‘97, June 1997, pp. 93-102, ISBN 3-211-83001-4.
[10] K. Dmitriev, S. Brabec, K. Myszkowski, H-P. Seidel, ”Interactive global illumination us-ing selective photon tracing”. In Thirteenth Eurographics Workshop on Rendering (2002), P. Debevec and S. Gibson, Eds.
[11] W. R. Mark, R. S. Glanville, K. Akeley, M. J. Kilgard, “Cg: a system for programming graphics hardware in a C-like language”. ACM Transactions on Graphics 22, 3.pp 896–907.
[12] F. E. Nicodemous, J. C. Richmond, J. J. Hsia, L. W. Ginsberg and T. Limperis, “Geometric considerations and nomenclature for reflectance. Monograph 161, National Bureau of Standards (US), Oct. 1977.

Towards a Parallel Image Mining System

J. Fernández, R. Guerrero, N. Miranda, F. Piccoli *

Líneas Informática Gráfica y Paralelismo y Distribución del
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 - San Luis - Argentina
e-mail: {*jmfer, rag, ncmiran, mpiccoli*}*@unsl.edu.ar*

Abstract

Images can reveal useful information to human users when are analyzed. The explosive growth in applying images as data in many fields of science, business, medicine, etc, demands greater processing power. With the advances in multimedia data acquisition and storage techniques, the need for automatically discovering knowledge from large image collections is becoming more and more relevant. Image mining, a relatively new and very promising field of investigation, tries to ease this problem proposing some solutions for the extraction of significant and potentially useful patterns from these tremendous data volume. This research field implies different stages, most of them demanding so many resources and computational time. The use of parallel computation is a good starting-point. Image mining process appears to be algorithmically complex requiring computing power levels that only parallel paradigms can provide in a timely way. As data sets involved are large, rapidly growing larger and images provide a natural source of parallelism, parallels computers could be organized to handle such big collection effectively. At this work we will examine the image mining problem with its computational cost, propose a possible global or local parallel solution and also identify some future research directions for image mining parallelism.

Keywords: Image mining, Image mining system, Parallel systems, Parallel techniques.

Resumen

El análisis de imágenes puede revelar información útil para los usuarios El significativo aumento del uso de imágenes en diferentes campos de la ciencia, medicina, negocios, etc., requiere de mayor poder de procesamiento. Con el avance en la adquisición de dato multimedial y de técnicas de almacenamiento, la necesidad de descubrir automaticamente conocimiento de grandes colecciones de imágenes aumenta. La minería de imágenes, area de investigación relativamente nueva y prometedora, trata de facilitar este trabajo proponiendo soluciones para la extracción de patrones significativos y potencialmente útiles a partir de grandes volumenes de datos. Comprende diferentes etapas demandantes de recursos y de tiempo computacional. El uso de computación paralela representa un buen punto de partida. El proceso de minería de imágenes parece ser algoritmicamente complejo, requiriendo niveles de poder computacional que solamente los paradigmas paralelos pueden proveer. Dado que involucra conjuntos de datos de rápido crecimiento y las imágenes representan una fuente natural de paralelismo, el paralelismo puede manejar semejante colección en forma efectiva. En este trabajo examinamos el problema de la minería de imágenes y su costo computacional, proponemos una possible solución global y local y definimos futuras extensiones para la minería de imágenes paralela.

Palabras claves: Minería de imágenes, Sistema de minería de imágenes, Sistemas paralelos, Técnicas de paralelismo.

*Grupo subvencionado por la UNSL y ANPCYT (Agencia Nac. para la Promoción de la Ciencia y Tec.)

1 INTRODUCTION

The tremendous growing of computerized information volume and variety has triggered the development of new data processing tools, World Wide Web technology and databases technologies that enable inferring useful knowledge from an important data bulk. As a result, it is necessary to support big collections of complex type information which includes complex objects data, spatial information or multimedia information. Many research works have focused on images and image mining.

The most general misinterpretation is that image mining only involves applying already existing data mining algorithms on images. Investigations in the area are usually pointed out into two main directions. The first one involves specific authority applications focusing on extracting most relevant image features, so they could be used in data mining [14][17][18]. The second direction applies to general applications, where the aim is discovering image patterns that might be useful in the understanding of existing interactions between human perception of the images at high level and image features at low level. Investigations in this direction try developments with major certainty of success in recovered images from a general purpose databases [13][20][24].

Human visual system has the ability to extract significant image relationships which are not represented in low-level primitive image features. Complex information and its use on specific applications leads to describe new association rules to information. The big challenge in image mining is extracting implicit knowledge, image data relationships, or other features not explicity stored in a pixel representation. As knowledge representation method, *patterns* have already been used by human being for simulating diverse cognitive process like intuition, intention and thinking. As long as the use of patterns can make the cognitive process more effective, they can be applied to describe the complexity and features of objects. Since the aim is to generate all significant patterns without any knowledge of the image content, diverse patterns types could be recognized: classification, description, correlation, temporal and spatial patterns.

Image mining deals with all aspects of large image databases including image storage, indexing schemes, and image retrieval, all concerning an image mining system [16]. Image databases containing raw image data as information, cannot be directly used for image mining purposes. Relational databases, traditionally used in data mining, do not satisfy this need; that is why other types of databases are defined like spatial, temporary, documentary and multimedia databases [26].

Figure 1 shows a general structure model for an image mining system. The system considers a specified sample of images as an input, whose image features are extracted to represent concisely the image content -Transformation and feature extraction phase-. Besides the relevance of this mining task, it is essential to consider invariance problem to some geometric transformations and robustness with respect to noise and other distortions while designing a feature extraction operator -Pre-processing phase-. After representing the image content, the *model description*

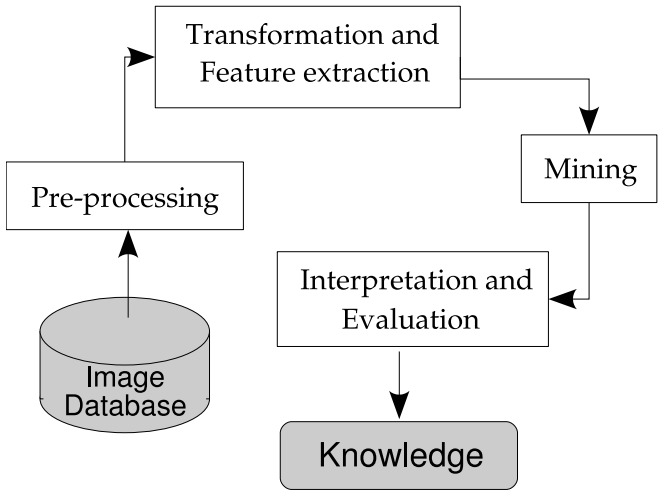


Figure 1: General Image Mining System

of a given image -the correct semantic image interpretation- is obtained. Mining results are obtained after matching the model description with its complementary *symbolic description*. The symbolic description might be just a feature or a set of features, a verbal description or phrase in order to identify a particular semantic.

The development of an image mining system is often a complex process since it implies joining different techniques ranging from data mining and pattern recognition up to image retrieval and indexing schemes. Besides, it is expected that a good image mining system provides users with an effective access into the image repository at the same time it recognizes data patterns and generates knowledge underneath image representation. Such system basically should assemble the following functions: image storage, image pre-processing, feature extraction, image indexing and retrieval and, pattern and knowledge discovery.

Image mining deals with the extraction of image patterns from a large collection of images, whereas the focus of computer vision and image processing is in understanding and/or extracting specific features from a single image. It might be thought that it is much related to content-based retrieval area, since both deals with large image collections. Nevertheless, image mining goes beyond the simple fact of recovering relevant images, the goal is the discovery of image patterns that are significant in a given collection of images. As a result, an image mining systems implies lots of tasks to be done in a regular time. Images provide a natural source of parallelism; so the use of parallelism in every or some mining tasks might be a good option to reduce the cost and overhead of the whole image mining process [1].

This works is structured as: the following section explains the differents difficulties and challenges involve in designing an image mining model. The section 3 explain the three main stages constituing a standard image mining system and their feasibility to be parallelize. Finally different parallel image mining models are proposed.

2 DIFFICULTIES AND CHALLENGES

Image mining deals with the study and development of new technologies that allow accomplishing this subject. A common mistake about image mining is identifying its scopes and limitations. Clearly it is different from computer vision and image processing areas. Moreover, the many knowledge discovery algorithms defined in the context of data mining are ill-suited for image mining. In image mining, there are many challengers still to overcome, some of them are:

- **Complexity of data:** To work with image and visual data is often to work with unstructured data, difficult to interpret and stored in a variety of different formats.

- **Scalability:** Image databases can easily reach hundreds of gigabytes and even terabytes in size. Scalable tools and algorithms for pre-processing and mining images that can manage such extremely large data in a reasonable time are yet to be developed. Massively parallel and high performance computing should help in this perspective for both image pre-processing and image mining.

- **Data inaccessibility:** Data adquisition and selection is fundamental in knowledge discovery process. The reasons for inaccessibility are multiple depending on the gathering means: sensors, satellites, among others. As incredible as it may seem, gathering images for research purpose or even industrial applications is not an easy task.

- **Privacy:** This has been an important issue with any data gathering and access. In some applications, image mining propels the problem of privacy a step further.
- **Minor support:** Image mining is relativity new, it relies heavily on fields such as vision and signal processing for data pre-processing and features extraction, fields which lack of adequate tool support, but in constant development.
- **Insufficient training:** Knowledge discovery and image mining tasks are related with many disciplines: artificial intelligence, databases, image and vision processing, high performance computing, visualization, etc. Interdisciplinary skills and work are required to process and cope with image.

To solve them in only one good application should be hard or impossible, but independent treatment of anyone could give notorious improvements to the whole image mining process.

3 PARALLELISM AND IMAGE MINING

Many issues of image mining can be optimized with different parallel techniques. Furthermore depending on tasks properties, different parallel paradigms could be applied in the same system. At a first glance, parallel applicant tasks will be: image storage, image processing, image indexing and retrieval and, pattern and knowledge discovery. In this section, the three main stages of an image mining system will be explained and then the feasibility of apply parallel paradigms at global and local level will be analized.

3.1 Processing Phase

Automatic image categorization involves experience on a real problem. The aim is to build a mining model using attributes extracted from and attached to the real problem, then evaluating the effectiveness of the model using new images. After the adquisition stage, the visual contents of the images in the database must be extracted and characterized by descriptive patterns - usually multidimensional feature vectors.

Orthogonal to challenges of developing specific image mining algorithms and models that operate on idiosyncrassy of images, one other major challenge for image mining is the pre-processing state previous to the extraction of relevant features. Generally, most of the images, if not all, are difficult to interpret, and a pre-processing phase is necessary to improve the quality of the images and make the feature extraction phase more reliable. The pre-processing state is arguably the most complex phase of the knowledge discovery process when dealing with images. If the pre-processing is well done, it can be decisive whether patterns could be discovered, or whether the discovered patterns could be interpreted at all. This phase often requires related expertise to computer vision, image processing, image interpretation, graphics and signal processing, domain knowledge or domain applications.

Pre-processing is always a necessity whenever the data to be mined is noisy, inconsistent or incomplete and it significantly improves the effectiveness of the data mining techniques. Nevertheless, a processing step is always done when applying any discovery technique for the extraction of relevant features. As a consequence, the process of building a mining model involves to split the processing phase into a pre-processing and extraction of visual features steps.

Figure 2 shows an overview of a categorization process. The first step is represented by the image acquisition and image enhancement, followed by feature extraction. The last one is the classification part, where different techniques for supervised learning are applied.

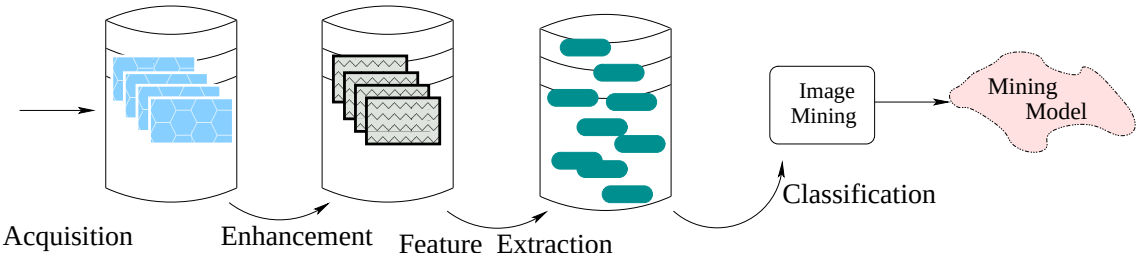


Figure 2: Image categorization process

The enhancement and feature extraction steps are usually refered as *Data Cleaning* and *Data Transformation* states that should be applied to the image collection. Data Cleaning is the process of cleaning the data by removing noise or other aspects that could mislead the actual mining process. Image enhancement helps in qualitative improvement of the image and can be done either in the spatial domain or in the frequency domain.

The most common techniques applied for data cleaning are the tipical image processing techniques like smoothing and sharpening filters. All these techniques could be combined with respect to a specific application.

On the other hand, data transformation implies to get an image content descriptor by means of its visual and semantic content. Visual content can be very general or domain specific. *General visual content* refers to color, texture, shape, spatial relationships, etc.; while *Domain specific visual content* is application dependent and may involve domain knowledge.

A good visual content descriptor should be invariant to any accidental variance introduced by the imaging process. A visual content descriptor can be either global or local. A global descriptor uses the visual features of the whole image, whereas a local descriptor uses the visual features of *regions* or *objects* to describe the image content. Moreover, as a previous step to obtain the local visual descriptors, an image is often divided into parts. The simplest way of dividing an image is to use a *partition*, which cuts the image into tiles of equal size and shape. A simple partition does not generate perceptually meaningful regions but is a way of representing the global features of the image at a finer resolution. A better method is to divide the image into homogenous regions according to some criterion using *region segmentation* algorithms that have been extensively investigated in computer vision. A more complex way of dividing an image, is to undertake a complete *object segmentation* to obtain semantically meaningful objects (like ball, car, horse).

Some widely used techniques for extracting color, texture, shape and spatial relationships from images are: Color Moments, Color Histograms, Color Coherence, Color Correlogram, Gabor Filter, Tamura features, Wavelet Transform, Moment Invariant, Turning Angles, among others [10][19].

Semantic content could be obtained by textual annotation or by complex inference procedures based on visual content. We will not discuss in detail this topic, trying to focus on our subject.

3.2 Mining Phase

In the nontrivial process of knowledge discovery in databases (*KDD*), data mining in general, and image mining in particular, have the aim of extracting implicit knowledge from data. They try to define valid, novel, potentially useful, and ultimately understandable patterns, relations or rules from them. These relations draw a *Predictive* or *Descriptive* model. With a predictive model is possible to estimate future or unknown values of interest, while with a descriptive model is possible to identify patterns which explain or summarize the analized data. Mining tasks depend in the model to be applied. *Classification* or *Regression* techniques define predictable models, while *Association Rule Mining* or *Clustering*, among others, define descriptive models. Image mining refers to a set of methods dedicated to the extraction of hidden knowledge from within an assortment of images. The early image miners have adopted existing machine learning and data mining techniques to mine for image information. Very few achievements have been realized and the approaches can be grouped in two classes. Those that discover patterns from:

- Images in large collections using the processed and extracted features within images;
- The image database using general descriptors.

While the applications vary from creating suitable models for image indexing to recognizing objects, categorizing images or image segments, the general tasks are similar and can be summarized as grouping images or features, either supervised or unsupervised, and associating image features.

The techniques frequently used include object recognition, image indexing and retrieval, image classification and clustering, association rules mining and neural network or a combination [8, 18].

3.3 Interpretation and Evaluation Phase

This task is a crucial one, it is tightly related with mining phase because it measures the quality from obtained patterns. Model preciseness can be secured by guarantying data independency between the training data set and testing data set.

Different evaluation techniques and measures can be applied. Evaluation measures could be objective or subjective. Which one of them would be used will depend on the mining taks to be done. The application context should be always considered when validating the obtained model [18, 13].

3.4 Global Parallel model

An image mining system(IMS) can be very computationally demanding due to the large amount of data to process, the response time required or the complexity of the involved image processing algorithms. Any parallel system requires dividing up the work so that processors can make useful progress toward a solution as fast as possible. The essential question is how to divide the labor.

There are three components to the work: computation, access to the data set, and communication among the processors [21]. These components are tightly related: dividing up the computation to make it faster creates more communication and often more data set accesses as well. Finding the best parallel algorithm requires carefully balance of the three named issues.

Parallelizing the image mining system showed at figure 1 involves to parallelize its three main areas: processing, mining and interpretation. Even though there exists a sequential line

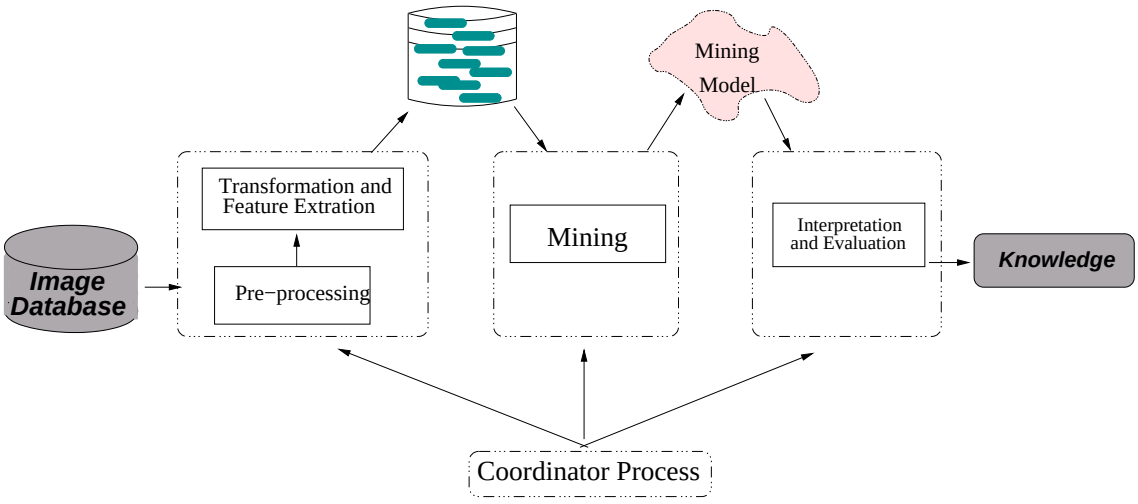


Figure 3: Global parallel arquitecture for the IMS

among them, it is a pseudo sequential line: after the first image set have been processed, the three named areas can be done in parallel.

Figure 3 shows the proposed global parallel arquitecture for the IMS. The model consists of four logical processing stages working in parallel: processing, mining, interpretation and coordination. The first three stages comes from the corresponding mining stages, and the last one from the derived management of the parallel model.

The coordinator process provides a GUI and task manager that directs the image mining process and is responsible for:

- At the starting of the mining process, it will coordinate the image mining tasks in a sequential way. First, the descriptor database generation through the processing phase, followed by the mining phase and finally the interpretation and evaluation phase.
- During the image mining process, it is responsible for the interaction with the image mining engine in terms of invoking, guiding and monitoring computations as well as visualization of the results.

Different parallel stage relationships are done by data sharing. Processing and mining stages share the feature database, and mining and interpretation stages share the mining model. Because reading and writing data structure accesses are simultaneous, synchonization mechanisms are required [25].

3.5 Local Parallel Model

At a refined level, each global parallel stage could be resolved in a parallel way. As a first attempt, we will focus only on the processing stage. This section sketchs three parallel levels concerning different parallel programming models and grains that could be accomplish collaboratively. The parallel alternatives are presented in an increasing complexity parallel order.

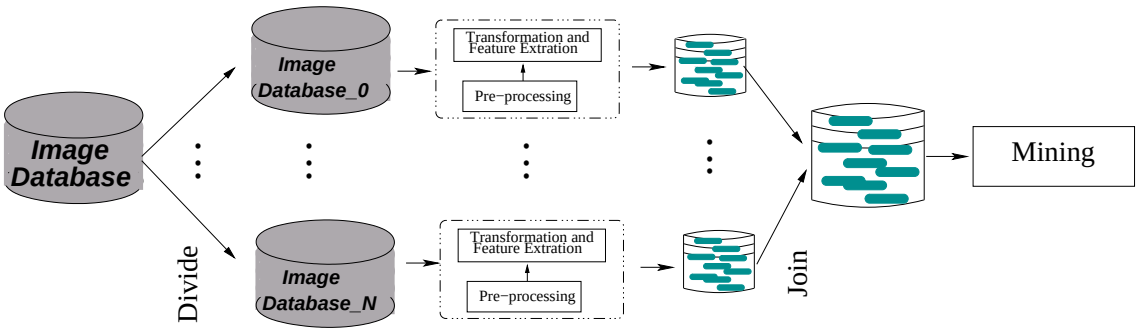


Figure 4: Level 1 Systems

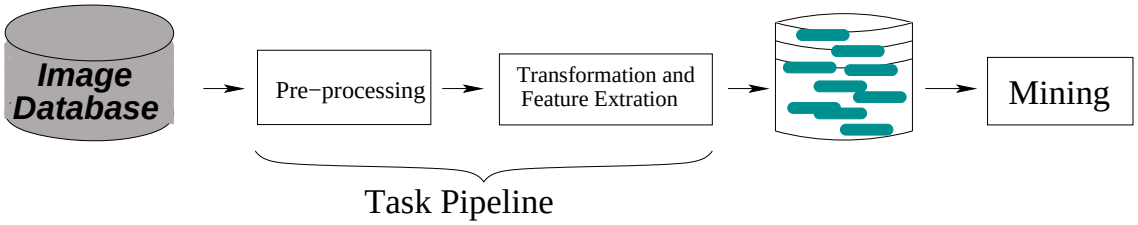


Figure 5: Level 2 Systems

3.5.1 Level 1: Embarrassingly Parallel

Applying the processing stage over the whole image set in the database could result in a time-expensive processing task. Besides this particularity, every internal data independence enables to draw a simple parallel model. The figure 4 shows the mentioned system architecture. N independent processes work on an image database partition, $DataBase\_i \quad (\forall i \quad 0 \leq i < N)$, making a feature data subset that will be joined to the whole working data set for the mining stage.

The parallel system has a coarse grain parallelism at data level following the $MDSP$ parallel programming model rules [2][15]. Moreover, as no particular effort is needed to segment the problem into a very large number of parallel tasks, and there is no essential dependency (or communication) between those parallel tasks, the problem is considered an embarrassingly parallel problem [25].

As each step can be computed independently from every other step, they could be made to run on a separate processor to achieve quicker results. An a-priori system performance estimation points out that it could be optimal or cuasi optimal.

3.5.2 Level 2: Parallelism into Processing Stage

At previous section only incoming system data independence was considered. At this section parallelism inside the processing stage will be take into account. Inside processing stage, as feature extraction step must be done after pre-processing, a pipelined processing is proposed, see figure 5 [25]. The pipeline has two well defined steps, the first one for image enhacement and the following for image feature extraction. As a consequence, a stream of images is passed through a succession of processes, each of which perform one task.

An interesting point to be considered in a pipelined parallel computation is the work de-

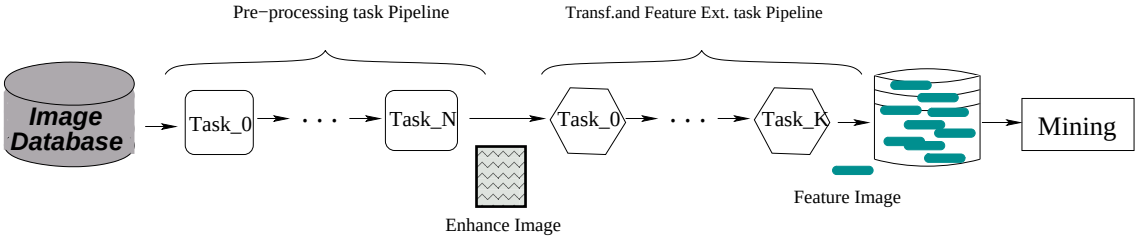


Figure 6: Refined Level 2 Systems

veloped into every step. When tasks workload is the same for every step, the pipeline gets the best performance. Either the enhancement or feature extraction steps address different processing task that should be done in an specific sequence. Dividing the enhancement or feature extraction steps (or both) into ordered substeps, could lead to take advantage from the inner step parallel characteristics and diminishing processing time problems due overwork into each of them. The system shown at figure 6 arises from the concepts stated. It can be observed a tasks pipeline where every task could be any classic processing task or a set of them, depending on the workload.

This last system has a finer parallel grain than the previous one where communications between tasks were increased.

3.5.3 Level 3: Parallelism Depending of Image Processing task

A sort of standard image processing tasks are commonly used at processing stage, like image smoothing, histogramming, 2-D FFT calculation, local area histogram equalization, local area, brightness and gain control, feature extraction, maximum likelihood classification, contextual statistical classification, image correlation (convolution, filtering), scene segmentation, clustering feature enhancement, rendering, etc. [6]. Many existing algorithmic implementions [3][5][7][9][11][12], could be done thru parallel solutions. Moreover, different techniques at different grain scale could be applied depending on the particular task; some of them are [4][22][23].

At this level any parallel model proposed not depends directly from the mining model itself, whereas it depends directly from any image processing task involved at the processing phase. As a consequence, any possible parallel model will be closely related to the specific image processing task to be done [10]; that is the reason because we do not suggest any model. The best solution could be to build a standard parallel image processing library that enables to make parallel processing at different combinations.

4 CONCLUSION

Integration of parallel tecniques into the image mining process was analized. It was considered from two points of view: global or local processing (each task into the global processing). Local analysis was focus on processing stage and three parallel image mining system models were proposed. They apply different parallel paradigm and grains: relations between comunication and computations.

At this moment, we are working on the parallel implementations of *level 2 system* as described in this paper. At the pipeline stage definition, the workload was considered. Moreover, the refined level 2 system is scalable and general. Scalability enables that any generic module

will be implemented, tested and assembled to the pipeline. Application domains other than image analysis will may also be benefited from the proposed methology. Generality enables to fit the pipeline states to the application context.

The current system has two stages implemented as a pipeline over a cluster of 15 nodes. The earlier results, besides they are few, are very promising.

REFERENCES

[1] H. Krawczyk A. Mazurkiewicz. A parallel environment for image data mining. In *Proceedings of the International Conference on Parallel Computing in Electrical Engineering (PARELEC'02)*, 2002.

[2] A.Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing.* Addison Wesley, 2003.

[3] D. Ballard and C. Brown. *Computer Vission.* Prentice Hall, Englewood Cliffs, 1982.

[4] J. Barbosa and J. Tavares A. Padilha. Parallel image processing system on a cluster of personal computers. *Lecture Notes In Computer Science*, pages 439 – 452, 2000.

[5] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing*, pages 433–481, 1993.

[6] A. Choudhary and S. Ranka. Parallel processing for computer vision and image understanding. *IEEE Computer*, 25(2):7–9, 1992.

[7] J. Crespo, J. Serra, and R. Schafer. Theoretical aspects of morphological filters by reconstruction. *Signal Processing*, 2(47):201–225, 1995.

[8] C. Djeraba. *Multimedia Mining, A highway to Intelligent Multimedia Documents.* Kluwer Academic Publishers, 2003.

[9] C. Giardina and E. Dougherty. *Morphological Methods in Image and Signal Processing.* Prentice Hall, 1988.

[10] R. Gonzalez and R. Woods. *Digital Image Processing, 2nd Edition.* Prentice Hall, 2002.

[11] B. Jahne. *Digital Image Processing: Concepts, Algorithms, and Scientific Applications.* Springer Verlag, 1997.

[12] A. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[13] Y. Keiji. Managing images: Generic image classification using visual knowledge on the web. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 167–176, November 2003.

[14] R. Kosala and H. Blockeel. Web mining research: a survey. *ACM SIGKDD Explorations Newsletter*, 2(1):1–15, June 2000.

[15] L.Yang and M. Guo. *High Performance Computing: Paradigm and Infrastructure.* Wiley-Interscience, 2006.

[16] R. Missaoui and R. Palenichka. Effective image and video mining: an overview of model-based approaches. In *MDM '05: Proceedings of the 6th international workshop on Multimedia data mining*, pages 43–52, New York, NY, USA, 2005. ACM Press.

[17] T. Mitchell, R. Hutchinson, M. Just, R.S. Niculescu, F. Pereira, and X. Wang. Classifying instantaneous cognitive states from fmri data. In *Proc. 2003 American Medical Informatics Association Annual Symposium*, pages 465–469, 2003.

[18] H. Orallo, R. Quintana, and F. Ramirez. *Introduccion a la Mineria de Datos*. Prentice Hall, 2004.

[19] J. Parker. *Algorithms for Image Processing and Computer Vision*. J. Wiley & Sons, 1997.

[20] A. Selim, K. Krzysztof, T. Carsten, and M. Giovanni. Interactive training of advanced classifiers for mining remote sensing image archives. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 773–782, 2004. (Industry/government track posters).

[21] D. Skillicorn. Strategies for parallel data mining. *IEEE Concurrency*, pages 26–35, 1999.

[22] W. Rapf M. ReinhardtL T. Braunl, S. Feyrer. *Parallel Image Processing*. Prentice Hall, Englewood Cliffs, Berlin Heidelberg, 2001.

[23] M. A. Vorontsov. Parallel image processing based on an evolution equation with anisotropic gain: integrated optoelectronic architectures. *Optical Society of America*, (16):1623–1637, 1999.

[24] Y. Wang, F. Makedon, J. Ford, L. Shen, and D. Goldin. Image and video digital libraries: Generating fuzzy semantic metadata describing spatial relations from images using the r-histogram. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 202–211, 2004.

[25] B. Wilkinson and M. Allen. *Parallel Programming: Tecniques and Aplications using Networked Workstations and Parallel Computers*. Prentice Hall, New Jersey, 1999.

[26] Ji Zhang, Wynne Hsu, and Mong Li Lee. Image mining: Trends and developments. *Journal of Intelligent Information Systems*, 19(1):7–23, 2002.

A Caminho de um Ambiente Adaptativo de Aprendizagem baseado no Nível de Aquisição de Conhecimentos do Estudante

Carlos E. Matsumoto, Danilo P. Martins, Felipe T. Carrasco, Leonel Gallo, Rodrigo S. H. Gonçalves, Edson P. Pimentel.

Universidade IMES
Av. Goiás, 3400, São Caetano do Sul, SP, Brasil, 09550-051

carlos.e.matsumoto@gmail.com, danilopio@gmail.com, felipe.carrasco@yahoo.com.br, leonel.gallo@gmail.com, r-hasperoy@uol.com.br, edson.pimentel@imes.edu.br

Abstract

Most of Learning Management Systems predominantly in use at computational support education are static and linear. They present the same instructional content for several students without considering their own knowledge acquisition level, becoming these environments sometimes unstimulating for students who are in a higher level and in other hand inhibited for students who are still in an initial level. This article aims to present the model of a computational learning environment based on the students' knowledge acquisition level. Providing to each student a suggestion of an instructional sequence suitable to their current profile, and dynamically updated, the environment will create conditions for students guide their own learning in a personalized way.

KEYWORDS: Personalization, Adaptive, Knowledge Acquisition Level.

Resumo

A grande maioria dos Sistemas de Gerenciamento de Aprendizagem predominantemente em uso no ensino com suporte computacional são estáticos e lineares, ou seja, apresentam o mesmo conteúdo instrucional para os diversos estudantes sem considerar o nível de aquisição de conhecimentos de cada um. Isso torna esses ambientes muitas vezes desestimulantes para estudantes que estão num nível mais avançado e inibidores para aqueles que estão num nível ainda inicial. Este artigo tem por objetivo apresentar o modelo de funcionamento de um ambiente computacional de aprendizagem baseado no nível de aquisição de conhecimentos do estudante. Ao disponibilizar para cada estudante uma sugestão de seqüência instrucional adequada ao seu perfil atual e atualizada dinamicamente, o ambiente criará condições para que os mesmos possam guiar o próprio aprendizado de forma personalizada.

PALAVRAS-CHAVE: Personalização, Adaptatividade, Nível de Aquisição de Conhecimentos.

1. INTRODUÇÃO

O crescimento explosivo da Internet, das tecnologias da comunicação e o acesso à informação criaram condições para expandir os horizontes da educação, com o uso de recursos computacionais,

para além da sala de aula presencial. Diversos recursos são utilizados para aperfeiçoar os métodos de ensino priorizando uma maior interação e participação do estudante no seu processo de aprendizagem. Para isso, os ambientes computacionais podem fazer uso de técnicas de inteligência artificial para personalizar as ações e reações do sistema ao perfil do aprendiz e também de mecanismos de comunicação síncrona entre professores e alunos.

No entanto, de acordo com Fernandes et. al. [1], a maioria desses ambientes trabalham com mecanismos estáticos que apresentam para todos os usuários o mesmo conteúdo. Isso torna esses ambientes muitas vezes desestimulantes para estudantes que estão num nível mais avançado e inibidores para aqueles que estão num nível ainda inicial.

Este artigo tem por objetivo apresentar o modelo de funcionamento de um ambiente computacional de aprendizagem baseado no nível de aquisição de conhecimentos do estudante. Ao disponibilizar para cada estudante uma sugestão de seqüência instrucional adequada ao seu perfil atual e atualizada dinamicamente, o ambiente criará condições para que os mesmos possam guiar o próprio aprendizado de forma personalizada.

O trabalho está organizado como segue: a próxima seção faz uma revisão dos conceitos de adaptatividade e apresenta exemplos de sistemas adaptativos. A seção 3 define nível de aquisição de conhecimentos e a seção 4 descreve o modelo de funcionamento de um ambiente computacional de aprendizagem baseado no nível de aquisição de conhecimentos do estudante. Finalmente, a seção 5 apresenta as considerações finais e as próximas etapas do trabalho.

2. SISTEMAS ADAPTATIVOS

Adaptatividade é usada para adaptar a visão de um usuário em relação a um determinado conteúdo a seus requisitos pessoais. Tecnologias de hipermídia são muitas vezes utilizadas para guiar um usuário através de um material digital, ajudando a compreensão do material. Existem muitas maneiras em que a adaptatividade pode ser implementada para ajudar o usuário [2].
Exemplos de sistemas Adaptativos:

a) *Adaptive Moodle:* O adaptive moodle propõe a integração de dois sistemas diferentes, chamados CMS do Moodle (Course Management System) e o APeLS (Adaptive Personalized eLearning Service) serviço de personalização adaptativa de eLearning. Com essa união foi possível combinar a robustez do CMS e a flexibilidade de um sistema adaptativo de hipermídia resultando em um sistema com suas melhores características, possibilitando ao aluno modificar o curso baseado nas suas principais necessidades [3];
b) *ELM-ART:* É um sistema educacional inteligente e interativo, cujo objetivo é auxiliar no aprendizado da linguagem de programação LISP. O ELM-ART contém todos os materiais de aprendizado on-line em formato de um livro texto interativo e adaptativo, o ELM-ART provê ajuda adaptativa, constrói a seqüência do curso individualizado, diagnostico da solução do estudante e ajuda na resolução de problemas baseando – se em exemplos [4].

De acordo com Brusilovsky et. al. [2], por hipermídia adaptativa entende-se: todos os sistemas de texto ou hipermídia que refletem algumas funcionalidades do usuário no modo usuário e aplica esse modelo para adaptar a vários aspectos visíveis do sistema para o próprio. Sistemas de adaptatividade de hipermídia constroem um modelo de objetivos, preferências e conhecimentos de usuário individual e utiliza isso através da interação para adaptar as necessidades do usuário.

3. NIVEL DE AQUISIÇÃO DE CONHECIMENTOS

A base para a adaptatividade em ambientes computacionais de aprendizagem é o perfil do estudante e também o modelo do domínio de conhecimentos. A figura 1 apresenta a arquitetura clássica de um Sistema de Tutoria Inteligente (STI) que possui uma estrutura onde cada módulo desempenha uma função específica dentro da arquitetura, além de estar inter-relacionado e em sincronia com os demais [5].

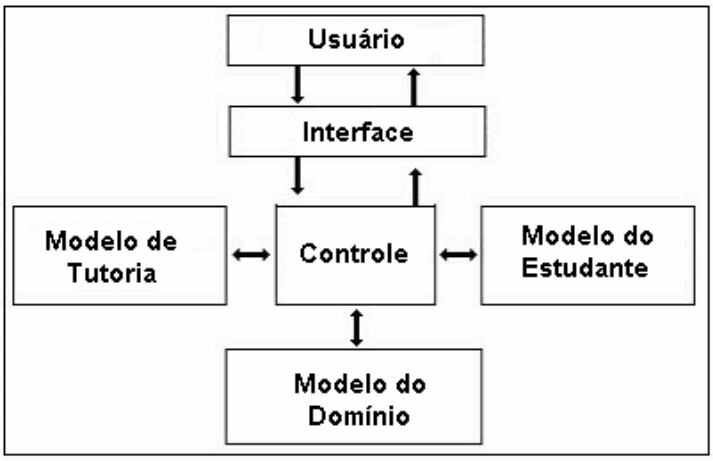


Figura 1 – Arquitetura Clássica de STI adaptada de Giraffa [5].

O Modelo de Domínio contém o domínio do conhecimento, que deve ser estruturado através de alguma técnica de representação do conhecimento. Neste trabalho adotou-se a estrutura na forma de hierarquia de conceitos organizados através de pré-requisitos de aprendizagem conforme demonstrado na figura 2.

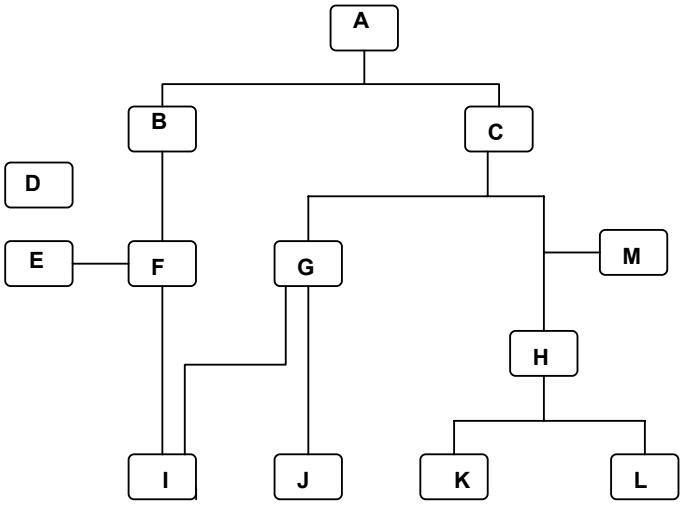


Figura 2 – Estrutura de Hierarquia de Conceitos

O Modelo do Estudante representa a crença do sistema sobre o grau de conhecimentos do estudante. Neste trabalho adotou-se como métrica cognitiva o Nível de Aquisição de Conhecimentos (NAC), conforme definido em Pimentel [6] como ”uma medida que indica o grau de conhecimentos do aprendiz em um determinado conteúdo, de um domínio de conhecimento, naquele instante”. O NAC pode ser obtido a partir de diversas atividades de avaliação do conhecimento e é composto de duas partes: índice de conhecimento e índice de avaliação. Índice de

conhecimento igual a zero e índice de avaliação igual a zero, indica que o conhecimento ainda não foi avaliado.

A tabela 1 apresenta a simulação do desempenho de um estudante em quarto tópicos, durante dez atividades de avaliação. Por exemplo, a linha 1 dessa tabela mostra que no "tópico 1", no instante T5, o estudante obteve uma medida igual a 0.8. O NAC instantâneo pode ser obtido através de media aritmética ou de média ponderada de acordo com o momento da realização da avaliação.

Tabela 1. Evolução do NAC em 10 avaliações.

TÓPICOS	**Avaliações – Time Line (0 to 9)**									
	T0	T1	T2	T3	T4	**T5**	T6	T7	T8	T9
Tópico 1	0.5	0.7	0.4	0.8	0.4	**0.8**	0.8	1.0	0.9	0.9
Tópico 2	0.2	0.5	0.5	0.4	0.5	0.5	0.7	0.6	0.7	0.8
Tópico 3	0.7	0.8	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0
Tópico 4	0.2	0.0	0.3	0.1	0.3	0.2	0.2	0.4	0.3	0.4

A próxima seção apresentará a proposta de uma arquitetura de um ambiente adaptativo de aprendizagem baseado no NAC do estudante em cada tópico ou conceito, e fazendo uso da estrutura hierárquica de organização dos conceitos.

4. ARQUITETURA DO AMBIENTE ADAPTATIVO

O Plug é dividido em dois módulos, de aluno e administrativo, cada um com tarefas distintas. A seguir serão detalhadas as funcionalidades que podem ser desempenhadas em cada um dos módulos.

4.1 Módulo Administrativo

O módulo administrativo é destinado aos professores e administradores do site, contendo as principais funcionalidades de um sistema de ensino virtual, como cadastro de cursos, alunos, professores e avaliações, que serão descritas a seguir.

4.1.1 Ambiente do Administrador

O Plug disponibilizará aos administradores as seguintes funcionalidades:

- Cadastro de cursos, definindo a categoria, tópicos, período de realização e de inscrições.
- Cadastro de usuários (administradores, professores e alunos) e suas permissões de acesso.

4.1.2 Ambiente do Professor

O Plug disponibilizará aos professores as seguintes funcionalidades:

- Inserção de conteúdo relacionado aos tópicos, que podem ser divididos em aulas.
- Definição das datas de realização das avaliações.
- Cadastro das questões que farão parte das avaliações ou questionários para adaptatividade, podendo ser dissertativas ou de múltipla-escolha. Para cada questão inserida, serão definidos os tópicos relacionados, bem como o peso para cada um.
- Correção das questões dissertativas, permitindo ao professor inserir comentários sobre a resposta e informar o aproveitamento do aluno por tópico.

- Relatório contendo o resultado das avaliações realizadas pelos alunos, mostrando o desempenho relativo a cada tópico.

4.2 Portal do Aluno

A seguir será detalhado o módulo de gerenciamento do conhecimento do aluno, de acordo com a Figura 3.

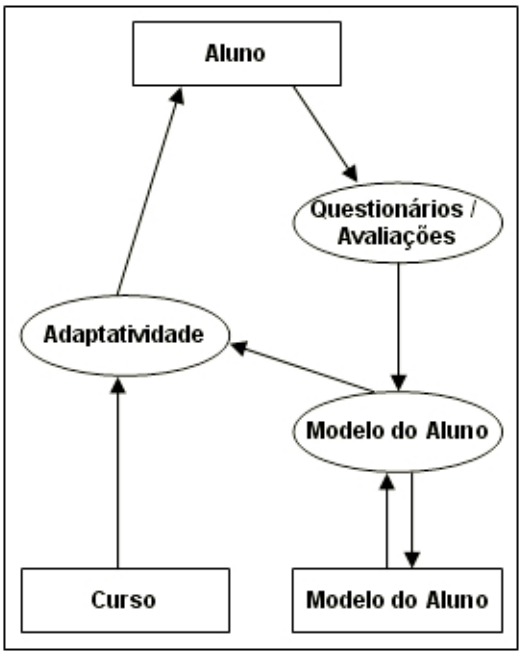


Figura 3 – Arquitetura do sistema.

- *Seleção do Curso pelo Aluno*: Ao informar o usuário e senha na tela principal do site será exibida uma listagem dos cursos habilitados para o aluno, definida pelo administrador (por exemplo, um professor). O aluno irá selecionar um dos cursos disponíveis e em seguida confirmar.

- *Verificação de Histórico*: No Plug os cursos são compostos por módulos (tópicos), que podem ser comuns a mais de um curso. Sendo assim, depois de selecionado o curso o sistema irá verificar em sua base de dados se o aluno já realizou alguns dos tópicos relacionados em cursos anteriores. Se o sistema identificar através do histórico do aluno que ele já cursou todos os tópicos referentes ao curso atual e teve um aproveitamento satisfatório, o sistema dará a opção ao aluno de realizar a avaliação referente a esse curso ou cursar novamente alguns módulos. Caso ele ainda não conheça todo o conteúdo do curso será exibida uma tela para seleção de conhecimento prévio.

- *Conhecimento Prévio*: O sistema disponibilizará todos os tópicos relacionados ao curso dando a opção do aluno selecionar os tópicos que ele já tem conhecimento, mas que não constam na base de dados do sistema. A Figura 4 ilustra a página para seleção dos tópicos. Em seguida será exibido um questionário para a avaliação destes tópicos.

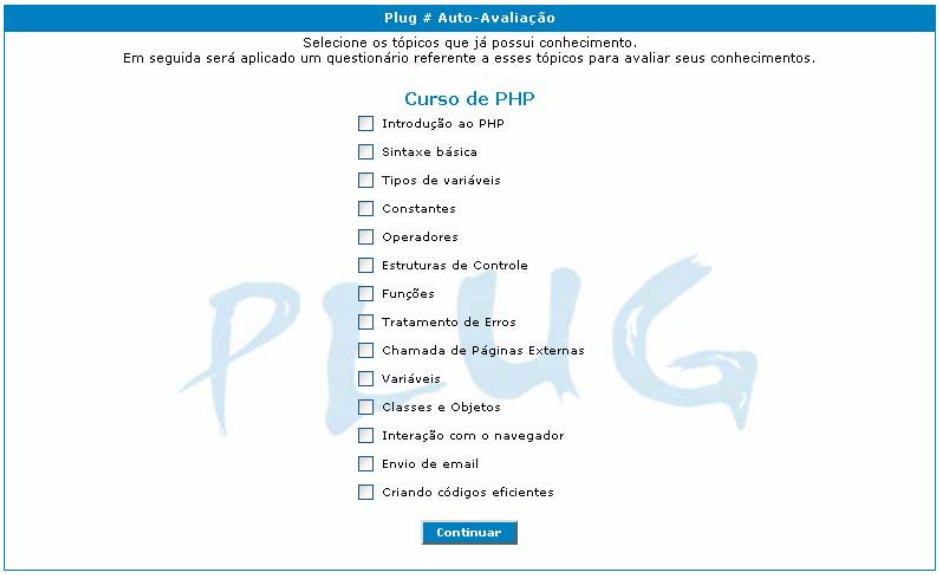


Figura 4 - Exemplo da tela de seleção do conhecimento prévio do aluno.

- *Questionário*: Será exibido um questionário de cada tópico que foi marcado pelo aluno. No final do questionário o aluno receberá um status de seu aproveitamento.
- *Curso Adaptado*: Na tela de curso adaptado serão listados todos os tópicos relacionados, onde os tópicos que o aluno já tem conhecimento estarão desmarcados, como mostrados na Figura 5. O conhecimento do aluno é verificado pelo sistema através do histórico e através do questionário realizado anteriormente. O aluno terá a opção de marcar os tópicos já cursados para realizar o estudo dos mesmos caso queira relembrar a matéria. Já os outros tópicos, onde o aluno ainda não tenha conhecimento, virão marcados e serão obrigatórios.

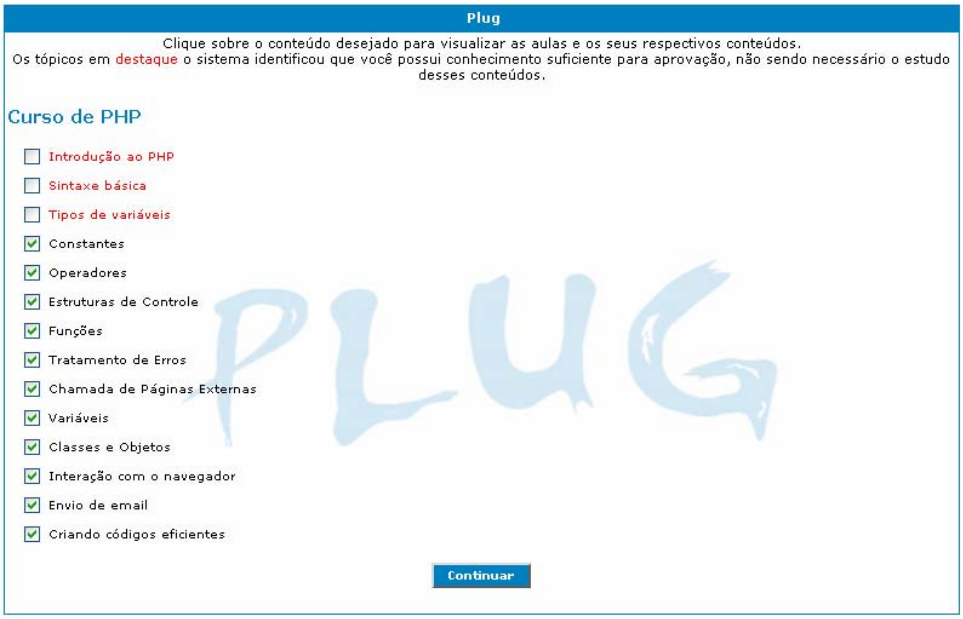


Figura 5. Exemplo da tela com os tópicos do curso adaptado.

- *Escolha do Tipo de Curso a ser Adaptado*: Após a seleção dos tópicos o aluno irá optar por realizar o curso adaptado ou realizar o curso completo (sem adaptatividade).
- *Execução Adaptativa*: O aluno deverá realizar o curso no tempo determinado pelo professor, seguindo a ordem dos tópicos que o sistema disponibilizará com base no conteúdo que foi adaptado.
- *Execução Completa*: O aluno deverá realizar o curso completo seguindo a ordem dos tópicos, sem adaptar nada, no tempo determinado pelo professor quando o curso foi cadastrado.
- *Avaliação*: A avaliação final será disponibilizada ao aluno após a conclusão do curso, no período especificado pelo professor.
- *Resultado*: Será exibido um relatório com seu aproveitamento por tópicos, e o resultado final do curso (aprovado ou reprovado).
- *Gravação de Histórico*: O sistema irá armazenar as informações de aproveitamento do aluno em relação a todos os tópicos para que no futuro isso possa ser utilizado para a adaptação de conteúdo para um novo curso.

4.3 Técnicas de Adaptatividade

As técnicas de adaptatividade utilizadas no Plug serão descritas a seguir.

4.3.1 Adaptatividade Histórica

Consiste em dar uma perspectiva de tempo ao aluno adaptando as representações de suas ações passadas no sistema. Diversas formas de representação são usadas como, por exemplo, trilhas ou rastros que são gerados pelo sistema. Essas representações podem ser customizadas pelo sistema com o objetivo de fornecer ao aluno uma visão histórica de suas ações [7]. No caso do Plug, a adaptatividade histórica será implementada da seguinte maneira: Depois de escolher o curso, o sistema verificará o histórico para que possa adaptar o curso ao aluno. Esse histórico conterá informações do aluno como notas, tópicos realizados, cursos, status, etc.

4.3.2 Conhecimento Prévio

Dependendo de seu conhecimento, os objetos de aprendizagem acessíveis a um aluno são determinados ao aplicar as meta informações sobre o relacionamento entre seus pré-requisitos entre os objetos de aprendizagem disponíveis e o conhecimento que é necessário para a sua compreensão. A adaptatividade ao estilo de comunicação do aluno significa a adaptatividade ao comportamento de uma comunicação do aluno como observado pelo sistema durante sua história de aprendizagem, [7]. A análise do conhecimento prévio do aluno será feita na verificação do histórico e na auto-avaliação. Na verificação do histórico o sistema verificará os módulos já realizados, notas, aproveitamentos e outras informações. Na auto-avaliação o sistema disponibilizará todos os tópicos relacionados ao curso dando a opção do aluno escolher o nível de conhecimento em cada tópico, mas que não constam no histórico do aluno.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Em grande parte ambientes de ensino virtual a adaptatividade de conteúdo não é implementada, não permitindo ao aluno o aproveitamento do conhecimento já adquirido.

O Plug através de gravação de histórico de aproveitamento do aluno em cursos realizados anteriormente possibilita a adaptatividade do conteúdo de acordo com o seu perfil. Mesmo que ainda não exista um histórico é possível o aluno indicar os tópicos referentes ao curso que já possui

um conhecimento prévio, possibilitando a adaptação do conteúdo. A estrutura hierárquica de conceitos também permite indicar ao estudante tópicos dependentes nos quais ele ainda não possui nível adequado para aprender um conceito mais complexo.

A próxima etapa deste trabalho será a realização de estudos de casos reais e a implementação de adaptatividade nas avaliações, permitindo além do conteúdo nivelado ao perfil do aluno, uma avaliação personalizada.

REFERÊNCIAS

[1] Fernandes, Silene de Freitas; Kienbaum, Germano de Souza; Guimarães, Lamartine N. F. Um Sistema Hipermídia Adaptativo para o Ensino de Simulação de Sistemas. III Workshop dos cursos de computação aplicada. São José dos Campos, SP, 2003.

[2] Brusilovsky, P. Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction. Carnegie Mellon University, Pittsburgh, 1996.

[3] Tiarnaigh, M. Adaptive Moodle. Trinity College, Dublin, 2005.

[4] Weber, G., Brusilovsky, P. ELM-ART: An Adaptive Versatile System for Web-based Instruction. International Journal of Artificial Intelligence in Education, Edinburgh, 2001.

[5] Pimentel, Edson P. Um modelo para avaliação e acompanhamento contínuo do nível de aquisição de conhecimentos do aprendiz. Tese de Doutorado. Instituto Tecnológico de Aeronáutica. São José dos Campos, SP, 2006.

[6] Giraffa, Lucia. "Uma arquitetura de tutor utilizando estados mentais". Tese de Doutorado, UFRGS, Porto Alegre, RS, 1999.

[7] Conlan, O. The Multi-Model, Metadata Driven Approach to Personalised eLearning Services. PhD thesis, Trinity College, Dublin, 2005.

Aplicación de Inteligencia Artificial para la Catedra Estructuras III

de la F.A.U. – U.N.N.E.

Congreso Argentino en Ciencias de la Computacion -CACIC 2007

Arquitecta y Docente Vanina Boccolini
I.T.D.A.Hu. (Instituto de Investigaciones Tecnológicas para el Diseño Ambiental del Hábitat Humano), Facultad de Arquitectura y Urbanismo, Universidad Nacional del Nordeste, 3400 Corrientes, Argentina, vnboccolini@arnet.com.ar

Resumen

El presente trabajo es una adaptación de la monografía presentada para la aprobación del Módulo "Representación, Almacenamiento y Recuperación del Conocimiento" para la Maestría en "Tecnología Informática Aplicada a la Educación" de la Facultad de Informática de la Universidad Nacional de La Plata durante el ciclo lectivo 2006.

En oportunidad de cursar dicho módulo pude conocer y comprobar los avances que en el campo de la **Inteligencia Artificial aplicada a la Educación** se están dando a través del estudio y experimentación de diferentes aplicaciones adaptadas específicamente contextos pedagógicos.

Es así que algunas herramientas informáticas del tipo de *Procesamiento de la Información, Sistemas Expertos* e incluso la *Evaluación en Línea* se convierten en recursos sumamente importantes para acrecentar y afinar resultados en las etapas del proceso enseñanza-aprendizaje de cualquier institución interesada.

La consigna de este trabajo fue *adaptar* algunas de estas herramientas al funcionamiento de una Cátedra de Nivel Avanzado de la Carrera de Arquitectura de la UNNE, que luego de muchos años de investigación y práctica en *innovaciones educativas aplicadas a la enseñanza de la Arquitectura,* finalmente han incorporado durante el ciclo lectivo 2006 la Educación Semipresencial como una nueva Oferta Académica Optativa para sus alumnos.

Palabras claves: Inteligencia Artificial, Educación a Distancia, Sistemas Expertos, Sistemas de Evaluación en Línea, Tecnología Educativa, Innovación Educativa.

1. SITUACIÓN ACTUAL

1.1. Justificación

El objetivo principal del presente trabajo monográfico es definir cual o cuáles aplicaciones de los tipos de Inteligencia Artificial son los más *convenientes* para adaptar al desarrollo de la cátedra Estructuras III.
Dicha aplicación persigue la finalidad de alcanzar una mejora del nivel educativo tanto de alumnos como de docentes, durante el proceso mismo de enseñanza-aprendizaje como en el momento de evaluación de resultados.

1.2. Descripción del Ámbito de Trabajo

La cátedra en la que soy docente es Estructuras III que funciona en el I.T.D.A.Hu. (Instituto de Investigaciones Tecnológicas para el Diseño Ambiental del Hábitat Humano) dependiente de la Facultad de Arquitectura y Urbanismo de la Universidad Nacional del Nordeste. El trabajo de docencia e investigación del cual soy auxiliar de investigación se centra en ***aplicar innovaciones educativas a la enseñanza de la Arquitectura*** utilizando, entre otras herramientas didácticas, la Tecnología Informática.
Las hipótesis planteadas en dicha investigación son verificadas de la experiencia directa que cada docente de las cátedras obtiene del contacto con los alumnos tanto en las aulas como en las tutorías (presenciales y vía correo electrónico). Los resultados obtenidos de esas experiencias, en los últimos 10 años, tienen como fin mejorar conceptual y experimentalmente el proceso de enseñanza-aprendizaje tanto de los alumnos como de los docentes.
Como ejemplo se puede mencionar el siguiente caso: se han hecho diagnósticos del manejo que los docentes hacen de la tecnología de multimedios para desarrollar una clase presencial y a partir de los resultados obtenidos se ha determinado cómo reaccionan los alumnos en cuanto al nivel que adquieren en sus propios procesos de aprendizaje.
Además se ha obtenido información del desenvolvimiento del docente frente a una clase presencial para poder luego optimizar sus habilidades y corregir sus desaciertos.
La incorporación de recursos como los multimedios y la comunicación a distancia con el alumno constituyen una propuesta superadora para los logros que se esperan obtener en la cátedra Estructuras III, siendo las actuales tutorías vía correo electrónico y el uso de la Plataforma Educativa CLAROLINE los medios alternativos para un vínculo significativo con los alumnos.
Es entonces en este marco de trabajo en el que se quiere experimentar con otras herramientas que, como ya se menciono, intervendrán en un proceso que abarca tanto el desarrollo lectivo de la cátedra como la evaluación final de resultados y por ende la reingeniería o reformulación permanente de la propuesta pedagógica implementada.

1.3. Diagnóstico de la Situación

De lo descripto se desprenden dos necesidades básicas que deberán ser satisfechas para el mejor funcionamiento de la cátedra. De aquí se puede empezar a dilucidar que tipo de ***inteligencia artificial*** sería la más adecuada a cada caso.
Por un lado se necesita adaptar nuevas herramientas ***alternativas de enseñanza-aprendizaje*** que contribuyan a perfeccionar la actual modalidad presencial de forma innovadora cumpliendo con el objetivo de la cátedra de sumarse definitivamente a la red de redes (www).
Por otro lado se necesita ***manejar el gran y variado volumen de información*** derivado de la administración propia de la cátedra, del trabajo del docente, de su relación con el alumno, de la producción propia del alumno, de la relación de los resultados producidos en la cátedra con el resto de la carrera de Arquitectura, etc., transformándolos en conocimiento útil para cumplir los fines de la investigación.

Para ambos casos es preciso explicar cómo se manejan actualmente las tres instancias del proceso de conocimiento en la cátedra:

1.3.1. Obtención de la Información.
Los **alumnos** son evaluados a través de su interacción (presencial y a distancia) con la cátedra, los exámenes presenciales parciales y finales (escritos y orales) y también en la entrega de un trabajo final de campo (análisis de un diseño estructural existente conocido internacionalmente).
Los tipos de datos obtenidos son de diferente naturaleza: el seguimiento de la *interacción* alumno-docente da cuenta del interés de ambas partes así como el grado de compromiso durante el cursado de la materia; *los exámenes* son evaluados con puntajes numéricos asignados a cada pregunta o ejercicio que contestan; el *trabajo final de campo* conlleva una evaluación mucho más subjetiva ya que los conocimientos que se manejan no solo tratan de cálculos estructurales estrictos sino valores completamente abstractos que derivan de la capacidad de análisis de situaciones que hace el alumno sobre una obra determinada en base a un autor y a un contexto específico. Finalmente todos los resultados son promediados en forma numérica (sumatoria de la evaluación diagnóstica, formativa, sumativa e interna del curso).
Los **docentes** también conllevan una evaluación completamente subjetiva ya que su forma de elaborar material y de presentarlo a los alumnos y su comunicación con ellos dependen principalmente del tipo de recursos formativos y los materiales con los que cuentan y de los objetivos establecidos por la cátedra.
A todos los datos obtenidos del desempeño de cada alumno y docente se suman la información de la que dispone la cátedra (material didáctico de muy diversos tipos tanto analógicos como digitales) para el desarrollo de sus actividades.

1.3.2. Almacenamiento de la Información.
Toda la información mencionada en el punto anterior deja claro la variada índole de la misma lo que dificulta su unificación tanto para el almacenaje como para la consulta de resultados.
Actualmente los resultados de seguimientos, exámenes y trabajos evaluados son volcados en planillas de cálculo y procesadores de texto. Respecto de los trabajos entregados por los alumnos todo el material didáctico se almacena en bibliotecas físicas (material analógico) y en CDs (material digital) ordenados e identificados por códigos en una base de datos realizada en planillas de cálculo.
Las planillas de seguimiento contienen valores de dos tipos: unos estrictamente numéricos o cuantitativos (de los que se pueden obtener estadísticas más precisas y gráficos demostrativos) y otros alfanuméricos que consideran el valor cualitativo de la observación y análisis del desempeño del docente y del alumno (a los que hay que asignarles un valor numérico estimativo para poder obtener las estadísticas y los gráficos mencionados).

1.3.3. Conversión, Recuperación y Consulta del Conocimiento.
El análisis de todos los datos descriptos se hacen: por un lado en forma manual visualizando los valores y estudiándolos en forma analógica; por otro lado se utilizan las herramientas de consulta y graficación disponibles en Excel para nuevamente visualizar los resultados.
Según lo anterior toda la información puede ser traducida en estadísticas pero para el caso de los valores altamente abstractos o cualitativos hay que forzarlos o traducirlos cuantitativamente para luego introducirlos al sistema de consulta (los resultados de tipo cualitativos son más bien elaborados en base a la lógica del evaluador y no de un ordenador o PC).
Se puede decir entonces que el conocimiento extraído de los datos obtenidos es manejado actualmente de manera obsoleta, ambigua e incompleta, lo que de ninguna manera significa que sean resultados equívocos o inservibles.

1.4. Objetivos Generales de la Aplicación de Inteligencia Artificial en Estructuras III

1 Implementar al funcionamiento de la cátedra nuevas tecnologías y técnicas que ayuden a optimizar los resultados educativos obtenidos tanto en docentes como alumnos.
2 Diseñar un nuevo proceso de trabajo que permita a la cátedra conjugar el actual modelo de enseñanza-aprendizaje con técnicas y herramientas avanzadas de trabajo on-line e inteligencia artificial aplicadas a la educación.
3 Diseñar un sistema de trabajo que permita a la cátedra generar una base de conocimiento única, útil y accesible para gestionar el conocimiento obtenido.

2. IMPLEMENTACIÓN DE INTELIGENCIA ARTIFICIAL

2.1. Objetivos Particulares de la Aplicación de I.A.

2.1.1. Personalizar la enseñanza para cada alumno de la cátedra sin interposición de límites de tiempo y/o recursos con una herramienta que a la vez de evaluadora actúe como de consulta y que almacene todo dato de la interacción con el usuario.
2.1.2. Identificar **patrones** de conductas similares de los alumnos ante sistemas de aprendizaje basados en web y de tipo de grado (convencional), con el fin de estudiar el proceso de aprendizaje que hace el alumno (incluyendo sus habilidades previas, el proceso de cambio durante el cursado de la materia, los resultados finales obtenidos).
2.1.3. Identificar **patrones** de conducta similares de los docentes a través de las técnicas de enseñanza (o técnicas didácticas) ante los dos sistemas educativos descriptos.

2.2. Hipótesis Iniciales

Luego de estudiar el material obtenido sobre *Inteligencia Artificial* (bibliografía básica del módulo y búsquedas en Internet) se pueden establecer dos hipótesis:

2.2.1. La aplicación de I.A. al trabajo realizado en la cátedra elegida no funciona creada ni aplicada aislada de las otras formas de trabajo de la misma. Esto significa acoplar el trabajo presencial con el no presencial sin descartar ni exaltar ni uno ni otro. Además trabajar con un equipo interdisciplinario donde los docentes expertos en el tema educativo sean apoyados por el trabajo de los especialistas en sistemas; y

2.2.2. El presente trabajo monográfico puede ser resuelto por los docente de la cátedra casi estrictamente desde el punto de vista de un ***usuario*** *entendido en el tema* limitándome a definir objetivos y fines de funcionamiento de un modelo informático y a partir de ello seleccionar los datos y otros elementos que le serán de utilidad al futuro *especialista en sistemas* que desarrollará la base de conocimiento y las aplicaciones necesarias.
Cómo se diseña el sistema y específicamente qué tipo de algoritmos son necesarios lo decidirá dicho experto ya que son demasiadas las posibilidades disponibles ya sean aplicadas en forma individual como agrupadas.
Como ejemplo de lo anterior basta mencionar que un algoritmo asociativo o uno de obtención de patrones fácilmente pueden trabajar conjuntamente con otro de tipo predictivo o genético. Pero ¿cómo definir acertadamente cuál utilizar? Eso es preferible que lo maneje el especialista en KDD.
Lo que ataña específicamente a la cátedra (docentes arquitectos e ingenieros) es estar al tanto de las posibilidades propuestas por la Inteligencia Artificial aplicables a la Educación y de ahí partir con la definición de los problemas que se quieren abarcar y que resultados se desea obtener.

2.3. Propuesta de Implementación de Herramientas de I.A.

Ya quedó claro que los objetivos perseguidos no solo plantean la necesidad de ***organizar*** la gran cantidad y variedad de **datos** operados en el curso de EIII sino también implementar nuevas herramientas informáticas que ayuden a ***perfeccionar el actual proceso de enseñanza aprendizaje*** de la misma.
O sea, por un lado *manejar información* de tipo cualitativa y cuantitativa y por otro lado utilizar *recursos educativos* que poseen directa influencia en el proceso de enseñar-aprender-evaluar. Todo esto aplicado a docentes, alumnos y al sistema administrativo y coordinador en general de la cátedra.
Se trata entonces de diseñar un sistema *iterativo y dinámico* híbrido que permita unificar el trabajo en paralelo entre:

- El trabajo en el aula o presencial con herramientas analógicas y digitales convencionales.
- El trabajo a distancia o semipresencial a través de la **Plataforma Educativa Claroline** (en uso) anexando un **SISTEMA EVALUADOR en LÍNEA** y un **SISTEMA EXPERTO,** y finalmente,
- Técnicas de **MINERÍA DE DATOS** que permitirán extraer todo el conocimiento necesario para cumplir con los objetivos propuestos previo almacenamiento preprocesado de ambas modalidades de trabajo.

A partir de ahora, con la inclusión de estas nuevas herramientas, el curso de Estructuras III se identifica como SISTEMA ESTRUCTURAS III y queda organizado según se describe a continuación (*figura 1).*

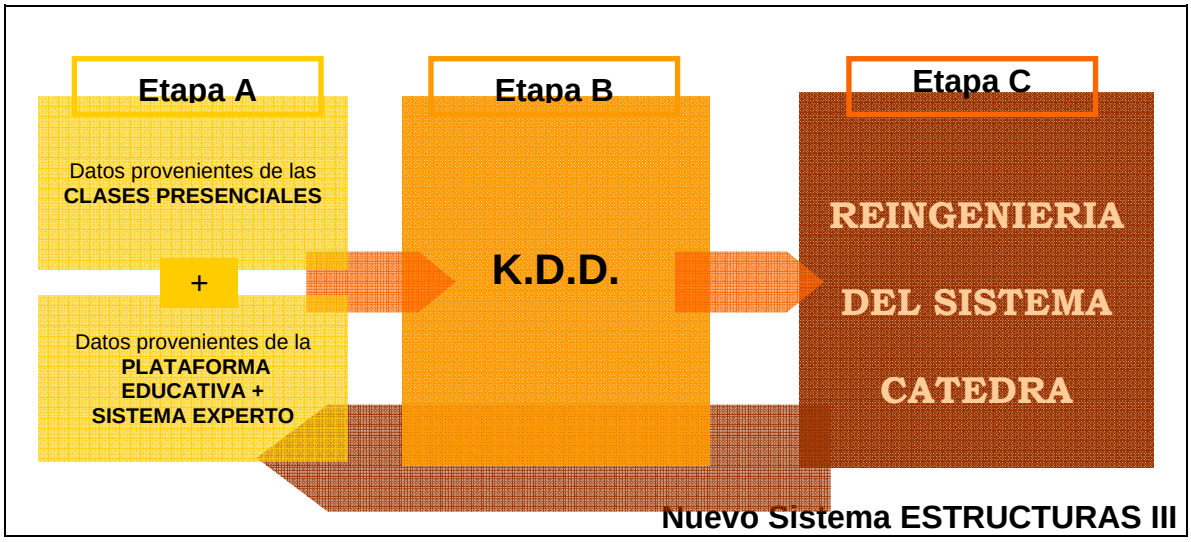


FIGURA 1. Nuevo Sistema ESTRUCTURAS III.

2.3.1. Modalidad de Cursado
BIMODAL (PRESENCIAL+SEMIPRESENCIAL)

2.3.2. Modalidad de Trabajo.

2.3.2.1. Clases Presenciales.
Significa presenciar un porcentaje del total del curso en las aulas de la sede de la facultad.
Utilidad de las Clases Presenciales: desarrollo de clases especiales del tipo Seminario-Taller de Diseño. Explicación de predimensionamiento de estructuras a través de trabajos prácticos. Debates de análisis estructural. Respuesta a consultas y defensa del Trabajo Final.
Como se explica en el siguiente punto, más adelante todo el material de estudio, de práctica y de evaluación se gestiona a través de la Plataforma Educativa.

2.3.2.2. Clases a Distancia o Semipresenciales.
Significa trabajar fuera de la sede de la facultad, en forma individual o grupal, según sea el caso, a través de la Plataforma Educativa Claroline. Esto no significa que el alumno necesite estar conectado a Internet el 100% del tiempo, sino solo el que le exija la tarea que este realizando (ej.: bajar material, hacer prácticas, conectarse con compañeros o docentes, etc.)
Utilidad de las Clases Semipresenciales: desarrollo de Seminario-Taller. Encuestas diagnósticas. Cuestionarios conceptuales. Obtención de material didáctico (apuntes digitalizados, ejemplos fotográficos, gráficos explicativos, presentaciones, videos multimediales y prácticas interactivas con laboratorios de medios), obtención y devolución de trabajos prácticos parciales y final, realización de consultas, prácticas y evaluaciones (auto-evaluaciones y evaluaciones parciales de avance), comunicación sincrónica o asincrónica con sus tutores y/u otros alumnos a través de las herramientas de comunicación.

2.3.3. Recursos Didácticos.

2.3.3.1. Clases Presenciales. Diseñadas especialmente para atraer la atención y el interés de los estudiantes y de los mismos docentes, estas cuentan con múltiples recursos audio visuales que convierten la típica clase de explicación de contenidos en algo dinámico e interactivo.
- Herramientas analógicas: discurso oral, grupos de discusión, pizarrón, sonido ambiente, retroproyector, puntero láser.
- Herramientas digitales: proyecciones multimediales (textos, imágenes fijas, videos, sonidos, laboratorios multimediales de práctica).

2.3.3.2. Clases a Distancia o Semipresenciales. Como recursos de trabajo a distancia se implementa el uso de una ***Plataforma Educativa***, existente, a la que se propone el linkeo a un Sistema de Evaluación Automático (T.O.L.) también existente y a un Sistema de Consultas que cumple la función de un **Sistema Experto de Consultas** que actualmente no existe (por lo que se propone su diseño e implementación en este trabajo).
La plataforma se denomina Claroline y fue diseñada y es administrada en el Politécnico de Turín (con quienes se trabaja en forma multilateral). La misma se puede visitar ingresando al sitio www.claroline.net
Al Sistema Experto se accede previa evaluación en línea que se hace a través de un Sistema de Evaluación Automático diseñado en el Politécnico de Turín especialmente para evaluación de sus alumnos en diferentes cursos. El mismo se denomina T.O.L. (Test on Line) y se accede a través de un linkeo desde nuestra plataforma educativa en una nueva sección que se denominará *Evaluación y Consultas*, sumándose así como nueva herramienta a demás de las ya mencionadas.

2.3.4. Gestión del Conocimiento.
Una vez alimentada la gran *Base de Datos*, eje neurálgico de la cátedra, se procede toda esa información para convertirla en conocimiento reutilizable que permita realizar una reingeniería de todo el sistema Estructuras III actualizando cada ciclo año a año.
La base de datos se alimenta de los datos cargados manualmente por docentes administradores y tutores del curso, los provenientes de la retroalimentación originados en la *plataforma educativa*, el *sistema de evaluación automática*, el *sistema experto*.
Para convertir ese gran volumen de información en conocimiento útil a la cátedra el mismo será procesado con técnicas de **K.D.D.** (Knowledge Discovery Data Base) cuyos resultados permitirán al plantel docente realizar la **Reingeniería** del todo el sistema.

2.4. Nuevo Esquema Funcional de Estructuras III + I.A.
De acuerdo a lo anterior a continuación se detalla la nueva estructura funcional de la cátedra como

sistema y se describe cada elemento que lo compone de acuerdo a la *figura 2*.

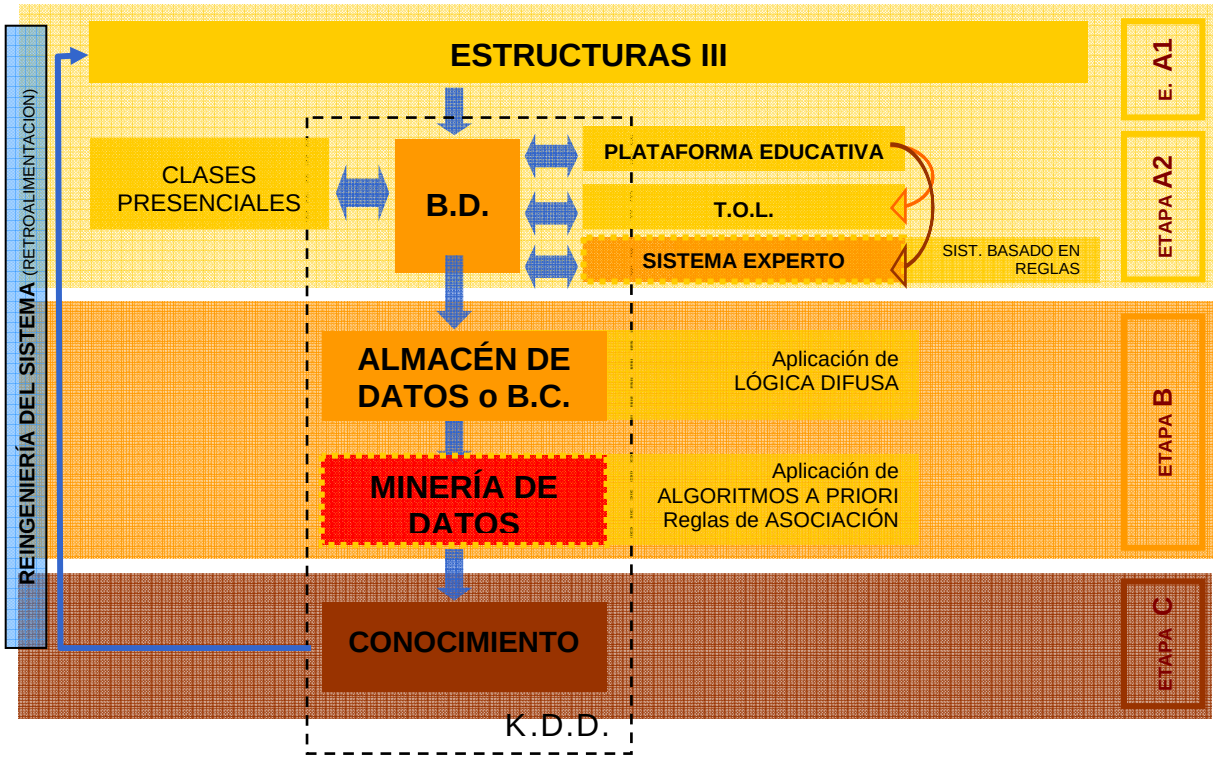


FIGURA 2. Diagrama funcional que representa mi propuesta para la generación de la nueva estructura de manejo del conocimiento en la cátedra Estructuras III habiendo incorporado al actual sistema presencial, el aprendizaje basado en web y las herramientas de Inteligencia Artificial investigadas.

2.4.1. Etapa A. Etapa de Enseñanza Aprendizaje
La misma se compone de dos niveles jerárquicos.

2.4.1.1. Etapa A.1. La Cátedra Estructuras III.
Esta primera etapa concentra la estructuración, administración y los recursos humanos y materiales necesarios para que la cátedra funcione. Diseño integral del funcionamiento de la cátedra. Espacio administrativo desde donde se toman todas las decisiones.

2.4.1.2. Etapa A.2. Materialización del Curso.
Este segundo nivel de la Etapa A contempla las herramientas necesarias para llevar adelante la ejecución del curso (*recursos humanos:* los docentes; *recursos materiales:* elementos didácticos, analógicos y digitales)
Es en este nivel donde aparece la primera implementación de Inteligencia Artificial a través de un Sistema Experto linkeado a la Plataforma Educativa.
A continuación se detalla en profundidad la materialización de cada herramienta componente de este nivel (ver *figura 2).*

2.4.1.2.1. Base de Datos.
Centro neurálgico de todo el nuevo Sistema Estructuras III. Almacena todos los datos referentes a docentes, alumnos y material de trabajo de la cátedra lo que genera una gran cantidad y variedad de tipos de archivos (xls, doc, pdf, mov, mpeg, jpg, psd, gif, etc)
Este gran almacén de datos se alimenta del ida y vuelta de toda la información manejada en este nivel ya que:

En una **primera instancia** se cargan todos los datos necesarios para que funcione la cátedra en un año lectivo específico.
En una **segunda instancia** la base de datos es retro-alimentada, en este nivel A, con la información devuelta por un lado de las clases presenciales (notas, apreciaciones, dificultades, etc.) y por otro de los sistemas informáticos utilizados (plataforma y sistema evaluador experto)

2.4.1.2.2. Clases Presenciales.
Ya se explicó en el punto 2.3 cómo funcionan y cuál es el fin de las mismas.
Lo que es necesario aclarar es que todo el contenido en estas expuesto es previamente diseñado, seleccionado y cargado por los docentes de la cátedra a la base de datos antes mencionada.
Con ese material ya previsto se eligen las herramientas analógicas y digitales que se usaran como medios de exposición.
2.4.1.2.3. Plataforma Educativa.
Esta contempla un modelo estándar diseñado por un grupo experto e interdisciplinario del Politécnico de Turín a la que se carga el curso específico que se desea gestionar.
La plataforma ofrece herramientas de comunicación entre docentes y alumnos, de seguimiento de las mismas, de exposición de novedades y de carga y descarga de materiales multimediales.
2.4.1.2.4. T.O.L. Test on Line
La cátedra actualmente está trabajando con investigadores del Politécnico de Turín, Turín, Italia en el marco de un proyecto ALFA Multilateral (Argentina, Italia, España, etc.) quienes hay desarrollado un Sistema de Evaluación Automático denominado **T.O.L. (Test on Line)**.
T.O.L. trabaja con el SQL Server de Microsoft como base de datos alimentada con preguntas y respuestas que conforman las pruebas que realiza el alumno todas tendientes a analizar los conocimientos del alumno y las habilidades y destrezas alcanzadas durante el curso.
Se trata de un cuestionario con respuestas cerradas entregado y realizado a distancia vía informática. Las preguntas surgen de un banco predefinido de preguntas y el sistema confecciona los cuestionarios alimentándose del mismo. La seguridad del sistema está definido por el criterio de tipo estadístico adoptado donde a más formulaciones de preguntas y respuestas para un mismo tema mayor posibilidad de configuraciones.
Terminada la evaluación el mismo T.O.L. confronta las preguntas del test con las respuestas asignando a cada punto un valor numérico que será el resultado de la evaluación. El alumno recibe en la Plataforma Virtual un reporte de los resultados y en base a ellos ingresa a interactuar con el Sistema Experto.
Caracteriza a este sistema evaluador la aparición de recursos multimediales en el proceso, según sea la necesidad, tal y como se propone también para el funcionamiento del sistema experto.

2.4.1.2.5. Sistema Experto.
Al tratarse de un curso dinámico con información que deberá ser actualizada año a año lo que se necesita es un sistema informático con características que permitan darle utilidad según las necesidades de la cátedra.
A diferencia de un sistema convencional los Sistemas Expertos:

- *Separan la información almacenada en una Base de Conocimientos de los procesos lógicos que ejecuta el sistema. Esto hace más fácil modificar por separado, según sea necesario, el conocimiento cargado en la base o los procesos programados.*
- *En el proceso de ejecución hay procedimientos que explican por qué se necesitan ciertos datos y cómo se llegó a determinados resultados. Esto permite un chequeo permanente del sistema y confiabilidad en el mismo.*
- *Para operar son más tolerantes ya que lo pueden hacer incluso con información desconocida.*
- *Trabaja con datos principalmente cualitativos.*

▪ *Captura, amplifica y distribuye el acceso a juicios basados en conocimientos y no necesariamente datos numéricos o textuales.*

El mecanismo sugerido para la implementación de estos nuevos sistemas es: una vez realizada la evaluación o autoevaluación con el T.O.L. los resultados son reportados en la Plataforma UNNE Virtual. Así el alumno queda en conocimiento de su rendimiento y es avisado o alertado de la necesidad de ingresar en la Herramienta de Evaluación y Consulta nuevamente pero esta vez para realizar una sesión con el Sistema Experto como tutor virtual, completando así el aprendizaje. Ello es despejar dudas y comprender conocimientos mal evaluados con el Sistema Experto como tutor virtual que lo guiará en el proceso.

Este sistema de trabajo de evaluación y tutoría individualizada permite a la cátedra discernir hasta que punto el alumno esta realmente aprendiendo y cual es el proceso mental realizado. Por ejemplo: ¿Por qué contestó de esta manera y no de esta otra? ¿Conocía y entendía el problema o fue tan solo suerte?

Además se automatiza todo el proceso de evaluación de resultados ahorrando tiempos de corrección de cada alumno y la posterior definición de notas.

El alumno e incluso docente que se preste a trabajar con el **Sistema Experto** estará interactuando con una máquina que no solo le solicitará que conteste preguntas sino que también le servirá de guía para entender claramente, en base a lo anterior, cuales fueron sus *debilidades y desaciertos* y trabajar sobre ello para completar sus conocimientos.

Esa interacción será individual (el sistema trabaja en base a los resultados que el alumno obtuvo de las evaluaciones con el T.O.L.), sin límites de tiempo, formulada expresamente para quien se está evaluando (personalización que ayuda al individuo y da mayor confiabilidad al sistema y contará con todo el material didáctico disponible en la base de datos de la cátedra para entender y aprehender a través de la práctica).

Ese espacio de tutoría donde el usuario se evalúa y al mismo tiempo consulta y aprende también tiene la función de guardar todo dato ingresado durante la interacción, lo que a su vez alimentará la base de conocimientos que será minada en una etapa posterior. Ello comprende todo tipo de dato que sea ingresado ya sea como texto, como imagen e incluso como parte de la interacción con un pequeño laboratorio de medios.

Como ya se mencionó este Sistema Experto no es parte de la Plataforma Educativa Claroline, sino que se linkea a través de ella dentro de un espacio denominado *Sección de Evaluación y Consultas.*

Este sistema experto trabaja Basado en Reglas. Esta estructura de tipo determinista es la más entendible para la lógica humana y es apropiada para la realización del presente trabajo.

En este caso el experto o especialista en sistemas, que carga la base de conocimiento, trabaja en conjunto con los docentes de la cátedra que definen las variables, las reglas y la relación entre ambas para que funcione el sistema.

2.4.2. Etapa B. Obtención de Conocimiento. Aplicación de Técnicas de Minería de Datos.

Es en esta etapa donde aparece la segunda aplicación elegida de Inteligencia Artificial: la **Minería de Datos**, que es a través de la que realmente se podrán cumplir con los objetivos propuestos sobre procesamiento de información, solo si las técnicas elegidas son apropiadas y suficientes.

Los pasos a seguir en esta etapa responden a la aplicación del concepto de KDD descriptos en adelante.

Se trata de un proceso que se inicia en la etapa A integrando de ella la base de datos como centro neurálgico del funcionamiento de la cátedra (*ver figura 2)*.

2.4.2.1. Comprensión del Dominio de la Aplicación.

Definición clara de los objetivos de diseño e implementación de un KDD. Identificación del problema a resolver.

La determinación de **Objetivos** específicos para este nivel permitirá avanzar a la siguiente etapa con normas preestablecidas. Aquí es donde es importante definir qué es lo que se quiere obtener del proceso de esta etapa. Exactamente que es lo que la cátedra desea extraer del proceso. A mayor definición menor incertidumbre y menos errores en el proceso que sigue.
Estos objetivos ya fueron enumerados en el punto 2.1. de este documento.

2.4.2.2. Diseño del Almacén de Datos.
Diseñar un esquema que unifique de manera operativa toda la información con la que se trabajará.
Se necesita para ello definir un **Modelo de Datos** (fuentes de datos, tipos de datos, su contenido, descripción y uso) que unifique y ordene el formato de los datos en la cantidad de bases de datos requeridas para que funcione el sistema. Esto es definir las variables que se van a minar.
Este Almacén de Datos se alimentará de los datos manejados en la BASE DE DATOS cargada con todo el contenido y datos que maneja la cátedra según se explicó en la etapa A. Lo que significa una amplia variedad de formatos que serán unificados a partir de la implementación del modelo de datos mencionado conformando así un almacén de datos con un formato de registros unificado.

2.4.2.3. Selección y Pre-Procesamiento de Datos.
Este paso significa suministrar los datos que se van a procesar. Primero definir los datos de entrada a explorar y analizar y luego su preparación previa unificación de formatos (previsto en el almacén de datos ya procesados), limpiando ruidos e información inútil, completando faltantes.
Ello implica desde identificar y seleccionar a preparar los datos que requiere el modelo de datos previamente diseñado. La preparación de los datos significa su limpieza, agregación, transformación y filtrado.
Es preferible realizar esta preparación antes de aplicar un análisis con minería de datos consolidando los datos de fuentes variadas para obtener un conjunto de registro de datos consistentes. A menor ruido, valores insuficientes, valores faltantes, etc., mejor.

2.4.2.4. Evaluación del Modelo de Datos.
Antes de aplicar la minería de datos y una vez cargada la base de datos según el modelo de datos elegido conviene inspeccionar como funciona el modelo de datos.
En este paso se pueden nominar las variables a permanecer y descartar aquellas que no serán útiles para el usuario final.
Esta validación incluso interviene en la toma de decisiones del tipo de algoritmos a utilizar en minería de datos y los parámetros de los mismos (como el soporte y el grado de confianza).

2.4.2.5. Elección y Aplicación de Técnicas de Minería de Datos.
Descubrimiento de relaciones, tendencias y trayectorias. Aplicado con fines exploratorios en información determinada, se busca posteriormente poder generar hipótesis de trabajo válidas.
Si bien esa es la forma de trabajo que se adopta al aplicar minería de datos, un procedimiento inverso al científico como lo describen los entendidos, en el caso particular de este trabajo el curso ya está diseñado y entonces la aplicación de minería de datos, más precisamente de KDD, tiene el fin de ajustar un modelo de curso existente validando o no su funcionamiento (reflejado en el procedimiento de los docentes y el aprendizaje de alumnos).
El éxito de esta tarea depende muy estrechamente de los pasos 1 y 4. Una vez definidos los algoritmos de procesamientos aplicables, paso 1, el sistema empezará a utilizar los datos obtenidos de la etapa anterior, paso 4.
Recién ahora se puede proceder a la extracción del conocimiento deseado.
Para el caso del presente curso el tipo de algoritmos a que se podrían utilizar en esta etapa son los de tipo ***no supervisados***, entre ellos los algoritmos **a priori de reglas de asociación** y

eventualmente la **Lógica Difusa** entre otros interesantes (**Web Usage Mining, Text Mining**)
A través del algoritmo a priori se busca descubrir reglas de asociación, es decir relaciones o afinidades entre conjuntos de ítems o variables expresadas en patrones que reflejen el comportamiento del alumno y del docente. Por lo tanto esas asociaciones tienen como fin descubrir patrones que permitan analizar un conocimiento útil a la cátedra según sus objetivos.
Un sistema interesante para ser aplicado es el INTELLIGENT MINER diseñado y desarrollado por IBM como un sistema completo que acompaña todo el proceso de KDD. Dicho sistema es actualmente utilizado en nuestra Universidad por el equipo de trabajo especialista en Inteligencia Artificial que funciona en la Carrera de Licenciatura en Sistemas en la Facultad de Ciencias Exactas, Naturales y Agrimensura. Este equipo ya ha hecho incluso análisis con datos provenientes de la misma Universidad para evaluar la calidad educativa de la misma.
Dado el caso ideal de trabajar con ellos como equipo interdisciplinario, sería más que obvio utilizar herramientas conocidas y ya incorporadas a su labor educativa e investigativa.

2.4.3. Etapa C. Aplicación del Conocimiento. EXTRAXIÓN DE RESULTADOS y su APLICACIÓN

2.4.3.1. Análisis de los Resultados.

Finalmente se pueden interpretar los datos descubiertos analizando los resultados obtenidos en la etapa anterior:

- Si el diseño e implementación del curso es correcto. Contenido repartido entre las clases presenciales y las semi-presenciales, recursos didácticos utilizados, tiempo y personalización dedicada a los alumnos, etc.
- Si el diseño e implementación de la evaluación es correcta. Tipo y formulación de los puntos, cantidad de preguntas, tiempo de respuesta, comportamiento del docente ante el examen, etc.
- Si el diseño del material didáctico y su aplicación en cada situación es el correcto. El formato es accesible a todos, es entendido por todos, su extensión es adecuada, el contenido es adecuado, etc.
- Si el tiempo de trabajo de los integrantes de la materia es el adecuado.
- Si la forma de trabajo de los integrantes de la materia es la adecuada.

2.4.3.2. Aplicación de los Resultados.

A partir de este momento se puede aplicar el conocimiento descubierto: retroalimentación del funcionamiento integral de la cátedra. Esto se traduce como la ***reingeniería del curso****.*
La primer aplicación de todo el sistema diseñado en este trabajo se hará con los datos actualmente manejados en la cátedra.
La siguiente aplicación y las sucesivas irán incorporando, descartando, arreglando los datos necesarios para que el sistema realmente demuestre un proceso evolutivo previsto como reingeniería del sistema.
Esto se produciría en ciclos en un espacio de tiempo definido entre la finalización de un año lectivo y el inicio del siguiente (el cursado de la cátedra es cuatrimestral):

Ciclo 1. Aplicación del Proyecto con los Datos existentes.
Ciclo 2. Aplicación del Proyecto + Reingeniería del Sistema.
Ciclo 3, Ciclo 5, Ciclo 6,……. Ciclo n.

3. CONCLUSIONES

En general los aspectos positivos y negativos de la implementación de la tecnología informática en el ámbito educativo están más que estudiados y explicados por autores especialistas en el tema tales como Edith Litwin y Beatriz Fainholc en nuestro pais.
Por ello en este último punto solo me limitaré a demostrar que tiene de positivo reorganizar la

cátedra Estructuras III según el esquema de funcionamiento desarrollado en esta monografía.
Para ello necesito dividir en dos partes estos comentarios. Por una lado los referidos a la **etapa de enseñanza – aprendizaje** del curso donde la tecnología informática interviene como herramienta didáctica. Y por otro lado la **etapa de gestión del conocimiento** donde se valida o no la etapa anterior.

Etapa de ENSEÑANZA – APRENDIZAJE.

En la cátedra se busca como principal objetivo que el alumno desarrolle sus capacidades de discernimiento y crítica ante el material educativo provisto por los docentes con el fin de lograr un conocimiento útil y racional que los capacite para resolver problemas de diseño de estructuras de grandes luces.
No basta con que el alumno comprenda y asimile contenido de la materia, sino que tiene que razonar lógicamente para resolver casos presentados hipotéticamente.
De allí es fácil comprender que para que el alumno alcance un conocimiento ideal es necesario no solo que *lea información y vea ejemplos* para luego ser evaluado, sino también que pueda interactuar con esa información para comprender mejor.
Por ejemplo trabajar con una simulación de física para comprobar experimentalmente que si una plegada triangular maneja ángulos muy agudos funciona mucho mejor y ahorra material que la misma estructura con ángulos mayores a 45º.
Obviamente que si, como está comprobado en las teorías de aprendizaje, el alumno puede ver la imagen de esa estructura y además modificarla, no fijará de memoria un concepto sino que lo asimilará rápidamente a través de la comprensión del fenómeno.
A ello hay que agregar que la interacción con un sistema experto no requiere de límites de tiempo, ni cantidad de consultas. Además todo lo que el usuario conteste o experimente queda registrado por el sistema para luego pasar a ser parte de la información que alimenta la base de datos que será minada.

En resumen el objetivo principal de la cátedra queda cumplido, ya que se puede lograr un seguimiento del proceso que realiza un alumno y no solamente registrar estadísticas de notas, cantidad de alumnos, cantidad de exámenes, etc.

De aquí surgen dos inconvenientes que deben ser previstos: 1. la formación del docente-tutor-guía ante esta nueva modalidad de trabajo y 2. el trabajo interdisciplinario entre docentes arquitectos e ingenieros, propios de la materia, los especialistas en educación y los especialistas en sistemas. Sin entrar en el tema de los escasos recursos económicos con los que contamos en cada Universidad Nacional.
Me faltaría agregar:

- La *agilidad de tiempos y recursos materiales* que se gana una vez pasada la primera etapa de diseño e implementación de todo el sistema cátedra nueva.
- *Mayor individualización del alumnado* teniendo en cuenta que cada curso lectivo cuenta con un promedio de 200 cursantes.
- Incentivo al estudio, la búsqueda, la creación y la crítica educativa gracias a la aplicación de las herramientas elegidas.
- La posibilidad de registro y seguimiento del proceso realizado por los mismos docentes.
- La disminución de errores.
- En general Se logran extraer conclusiones más completas posibilitando mejores resultados pedagógicos en alumnos y docentes.

Etapa de GESTIÓN del CONOCIMIENTO.

Dado el gran y variado volumen de situaciones e información que se presentan en el desarrollo del curso, está de más afirmar que aplicar técnicas de tratamiento, manejo y extracción de conocimiento son necesarias en el proceso.

Ello se refleja en le mayor seguridad y menor margen de errores que un sistema informatizado presenta en comparación con el análisis manual de datos como se hacen en la actualidad.
El tratamiento manual de los datos y las conclusiones realizadas en base a la experiencia de una persona es una tarea muy laboriosa que exige mucho tiempo físico y resulta en conclusiones incompletas y datos mal almacenados.
La aplicaciones de las técnicas de K.D.D. disminuyen los tiempos físicos del proceso que realiza el humano, consigue extraer no solo la información solicitada sino también encontrar información oculta y además permite que toda la información sea almacenada de manera tal que todo el proceso puede sufrir un cambio, pequeño o considerable, de manera automática sin invalidad todo el funcionamiento.
Así, la *reingeniería* del sistema cátedra Estructuras III es posible sin tener que hacer reestructuraciones completas sino solo atacar o arreglar los puntos débiles sin que esto invalide el resto.
Además la información almacenada es útil para otros usos, internos o externos, puede ser ampliada sin problemas.

4. BIBLIOGRAFÍA

Los datos bibliográficos se seleccionaron de acuerdo al tipo de aplicación para las que fueron creadas, el medio en el que se aplicaron y teniendo en cuenta la credibilidad de las bases bibliográficas por ellos usadas. En general son trabajos de investigación aplicados específicamente a educación y se relacionan directamente con el tema de este trabajo por lo que son bibliografía específica toda ubicable en Internet.

- "*Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza basada en web*". Cristóbal Romero Morales, Sebastián Ventura Soto, Cesar Hervás Martínez. Departamento de Informática y Análisis Numérico. Escuela Politécnica Superior. Universidad de Córdoba. España. Año 2005.
- "*Descubrimiento de Reglas de Predicción en Sistemas de e-learning utilizando Programación Genética*". Cristóbal Romero, Sebastián Ventura, Cesar Hervás. Universidad de Córdoba. Campus Universitario de Rabanales. Córdoba. España. Año 2005.
- *"Evaluación del aprendizaje en física: cinco años de experimentación en el Politécnico de Turín". Mateo Luca Ruggiero, Angelo Tartaglia, Elena Tresso. Departamento de Física del Politécnico de Turín. Italia. Año 2005.*
- *"Minería de Datos. Trabajo monográfico de adscripción."*. María Inés Kubski. Director Mgter. David Luís la Red Martínez. Cátedra Diseño y Administración de Datos de la Carrera Licenciatura en Sistemas de Información. Facultad de Ciencias Exactas, Naturales y Agrimensura, Universidad Nacional del Nordeste. Corrientes, Argentina. Año 2004.
- *"Minería de Datos con Intelligent Miner. Trabajo monográfico de adscripción."*. María Inés Kubski. Director Mgter. David Luís la Red Martínez. Cátedra Diseño y Administración de Datos de la Carrera Licenciatura en Sistemas de Información. Facultad de Ciencias Exactas, Naturales y Agrimensura, Universidad Nacional del Nordeste. Corrientes, Argentina. Año 2004.
- *"Monografía sobre Sistemas Expertos"*. Bargiela Roberto Alejandro. Cátedra Sistemas Operativos, Departamento de Informática, Carrera de Licenciatura en Sistemas de Información, Facultad de Ciencias Exactas y Naturales y Agrimensura, Universidad Nacional del Nordeste. Corrientes, Argentina. Año 2003.
- *"Inteligencia Artificial. Sistemas Expertos. Redes Neuronales"*. L. Arrúa y E. Meza Fernández. Carrera Licenciatura en Sistemas de Información. Facultad de Ciencias Exactas, Naturales y Agrimensura, Universidad Nacional del Nordeste. Corrientes, Argentina. Año 2003.

- *"Tema II: Proceso de desarrollo de minería de datos"*. Materia: Minería de Datos. Facultad de Ciencias, Escuela de Computación, Centro de Computación Paralela y Distribuida, Laboratorio de Inteligencia Artificial, Universidad Central de Venezuela. Año 2003.
- *"Aplicación de técnicas de minería de datos en la construcción y validación de modelos predictivos y asociativos a partir de especificaciones de requisitos de software"*. María N. Moreno García, Luís A. Miguel Quintales, Francisco J. García Peñalvo y M. José Polo Martín. Universidad de Salamanca. Departamento de Informática y Automática. España. Año 2001.

ColoquiosEC, un proyecto software para el seguimiento de alumnos universitarios iniciales

Marcela Fabia López
Facultad de Ciencias Exactas, Universidad Nacional de Salta,
4400 Salta, Argentina,
mfflopez@unsa.edu.ar

Marcia Ivonne Mac Gaul
Facultad de Ciencias Exactas, Universidad Nacional de Salta,
4400 Salta, Argentina,
mmacgaul@cidia.unsa.edu.ar

María Laura Massé Palermo
Facultad de Ciencias Exactas, Universidad Nacional de Salta,
4400 Salta, Argentina,
mlmassep@gmail.com

Eduardo Francisco Fernández
Facultad de Ciencias Exactas, Universidad Nacional de Salta,
4400 Salta, Argentina,
effer@cidia.unsa.edu.ar

Ariel Fernando Liendro
Facultad de Ciencias Exactas, Universidad Nacional de Salta,
4400 Salta, Argentina,
afliendro@gmail.com

Resumen

En este trabajo comentaremos los resultados obtenidos a partir de la experiencia de diseño y desarrollo de un software de evaluación, denominado *ColoquiosEC* y su aplicación, durante los años 2006 y 2007, al proceso de aprendizaje de los alumnos de primer año universitario de la carrera Licenciatura en Análisis de Sistemas de la Universidad Nacional de Salta, la cual cuenta con una matrícula promedio anual de quinientos alumnos, de los cuales un setenta por ciento llegan a las primeras instancias de evaluación. La experiencia de incorporación de este producto es analizada desde tres perspectivas. La primera centrada en el propio proceso evaluativo, tendiente a medir los aprendizajes del alumno, la segunda, indagando sobre la acción formativa y sus adecuaciones metodológicas y por último, la evaluación del propio material educativo tomando como fuente principal las opiniones y sugerencias del estudiantado.

Palabras claves: NTICs, Software Educativo, Evaluación, Ingreso Universitario

Abstract

In this work we will comment the results obtained from the design experience and development of a software of evaluation, denominated *ColoquiosEC* and its application, during years 2006 and 2007, to the learning process of beginners university students of Systems Analyst carreer, National University of Salta, which counts on a matriculation annual average of five hundred students, from who a seventy percent arrives at the first instances of evaluation. The experience is analyzed from three differents perspectives. First one, centered in the evaluation process itself, wich objetive is to measure the learnings of the student, second, investigating on the formative action and its methodologic adjustments and finally, the evaluation of the educative material taking like main source the opinions and suggestions from the students.

Keywords: NTICs, Educative Software, Evaluation, Beginners University Students

1. INTRODUCCIÓN

La formación integral del ser humano en cuanto a su capacitación científica, social y tecnológica como también la transmisión de valores y de cultura es lo que la sociedad requiere de las instituciones de enseñanza superior. Las universidades han dado respuesta a esta necesidad construyendo sus bases en la modalidad presencial de su oferta académica.

Por otro lado, en los últimos tiempos, la modalidad de educación a distancia tomó mayor auge gracias a la incorporación de la multimedia y de nuevas tecnologías de comunicación

La utilización de estas nuevas tecnologías permiten acortar las distancias geográficas a través de las redes de comunicación y dar respuesta a una de las principales problemáticas de la educación a distancia: la interactividad. Estas ventajas también pueden ser aprovechadas en la educación presencial, en particular en aquellos ambientes en los que el alumnado numeroso impide una fluida interacción con el docente. En este sentido, las nuevas tecnologías pueden intervenir en el proceso de aprendizaje jugando un rol facilitador de la comunicación entre los actores de un proceso de aprendizaje, tal como lo manifiestan Briones y Martínez, "Interesa recuperar los espacios de interacciones próximas, tanto reales como virtuales, que favorezcan el encuentro entre protagonistas de la educación a distancia como productores de sentido" [1].

Desde el año 2002, la cátedra de Elementos de Computación, de la carrera de Licenciatura en Análisis de Sistemas de la Universidad Nacional de Salta, inició el desafío de incorporar el uso de recursos tecnológicos, en particular informáticos, al servicio del aprendizaje de los alumnos. Resultó imprescindible partir de algunas preguntas claves al momento de iniciar esta propuesta: cuáles y cómo son los estudiantes para los que se realizará la actividad, qué y para qué se busca que el estudiante aprenda, cómo puede lograrse ese aprendizaje y cómo evaluar lo adquirido.

Dicho de otro modo, cuáles son las características socioculturales del grupo de estudiantes y cual es la propuesta a implementar, atentos a su finalidad formativa y a los contenidos que se pretende transmitir a los estudiantes. La programación, ordenamiento, organización de espacios, etc. no puede ser azaroso sino adecuarse a estas especificaciones. Factor fundamental es el cómo, es decir las cuestiones vinculadas con la metodología, estrategias y procedimientos propios del rol docente. Finalmente, la tarea de evaluación por la que se produce la verificación del proceso y de los resultados, será el indicador último que posibilite reencausar la propuesta para ajustarla a las particularidades de cada situación de aprendizaje

Por lo tanto, entendemos que la tecnología debe estar al servicio de la educación, como herramienta que contribuya a la acción formativa, cuya principal función es la de alcanzar los objetivos propuestos. En este aspecto es que reconocemos la función fundamental de la evaluación y nos planteamos el lugar que ocupa la tecnología en esta instancia. Por otra parte, dependiendo de su funcionalidad distinguimos otras formas de evaluación aparte de aquella que permite medir los aprendizajes de los alumnos. Interesan también aquellas, no menos importantes, dirigidas a evaluar la propia acción formativa y consecuentemente, la de los materiales educativos desarrollados sobre un determinado soporte tecnológico.

En este trabajo comentaremos los resultados obtenidos a partir de la experiencia de diseño, desarrollo e implementación de un software de evaluación y su aplicación al proceso de aprendizaje de los alumnos de primer año universitario que cursan la asignatura Elementos de Computación, la cual cuenta con una matrícula promedio anual de seiscientos alumnos, de los cuales un setenta por ciento llegan a las primeras instancias de evaluación.

Este programa, denominado *ColoquiosEC*, ha sido desarrollado por especialistas informáticos y en educación que a su vez son docentes de la materia. Se integró desde el año 2006 a los otros programas de software educativo elaborados a medida. Al igual que ellos, *ColoquiosEC* surge como un producto desarrollado por un grupo de I+D denominado PI (Productos Interactivos). 

2. LAS NUEVAS TECNOLOGÍAS EN APOYO DEL INGRESO UNIVERSITARIO MASIVO

La experiencia con un alumnado numeroso que inicia sus estudios universitarios nos indica que las estrategias metodológicas de la modalidad presencial no siempre son las más adecuadas para esta población estudiantil, caracterizada por jóvenes que asisten a la universidad pero no participan activamente de las instancias tradicionales de esta modalidad. Por ello, asumimos el desafío de democratizar la educación, abriendo nuevas oportunidades de acercamiento a alumnos, que aún asistiendo a la universidad bajo una modalidad presencial, su aprendizaje es a distancia.

Como plantea Fainholc "Educación a Distancia no es sólo la separación física de educadores y estudiantes en la consumación del acto de aprender. Existen otros tipos de separaciones que deben ser tenidas en cuenta a la hora de discutir el tema de la interactividad. Se trata de separaciones psicológicas, sociales, culturales y económicas, con lo que expandimos y complicamos la definición clásica de enseñanza a distancia" [2].

En este sentido, también cabe hacernos la siguiente pregunta: ¿tiene nuestra sociedad local y en particular los ingresantes a esta carrera universitaria la madurez suficiente como para ser partícipes activos de su proceso de aprendizaje, atentos a que se requiere de compromiso, iniciativa, interacción de grupos, trabajo interdisciplinario, trabajo colaborativo y fundamentalmente la modificación de su conducta en lo que respecta al seguimiento y control reflexivo de su aprendizaje? Se requiere de la modificación de las conductas personales de todos los actores participantes del proceso de enseñanza y aprendizaje, dirigidas a formar comportamientos responsables y comprometidos para lograr un aprendizaje eficaz.

Con el principal objetivo de paliar esta problemática, se viene trabajando sistemáticamente en la incorporación de las NTICs en la primera asignatura de la carrera. Se han diseñado e implementado materiales educativos en soporte informático: *EC_Soft* y *SisNum*, que asisten al alumno en sus aprendizajes, desde la presentación de los contenidos hasta las instancias de auto evaluación.

Las conclusiones de los primeros tres años indican no sólo una mejor relación recursos-cátedra, tanto humanos como físicos, temporales y tecnológicos, sino que, fundamentalmente se observan los primeros indicios de adecuación a las nuevas estrategias metodológicas adoptadas. Procuramos que estas nuevas estrategias permitan al alumno presencial tomar distancia de los mecanismos tradicionales de estudio y fortalecer un aprendizaje más autónomo.

Entendemos que, cuando se usan NTICs debe fortalecerse el seguimiento del alumnado que ahora trabaja más autónomamente. Nuestras estadísticas sobre cantidad y propósito de acceso a los programas *EC_Soft* y *SisNum* indican que los mismos están siendo aprovechados, sobre todo, para instancias de auto evaluación y ejercitación. Simultaneo a la incorporación de estos materiales educativos informatizados, la cátedra implementó evaluaciones rápidas (coloquios) para medir semanalmente el nivel de los aprendizajes, lo cual impactó favorablemente en los alumnos, quienes reconocieron su importancia en la definición del nivel de avance alcanzado. Hasta el año 2005, estas instancias evaluativas se desarrollaban exclusivamente en forma presencial, asignando para su

resolución un breve intervalo de tiempo. La incorporación del producto *ColoquiosEC* se produce a partir del año 2006 y se prosiguió en el 2007, como una prueba piloto aplicada a una muestra aleatoria de cincuenta alumnos en cada periodo lectivo, como un primer paso para el estudio de la experiencia y su posible transferencia a instancias de evaluación a través de Internet. Es así que en durante esta experiencia, convivieron la modalidad tradicional de evaluación de coloquios y la implementada para el grupo piloto, quienes también debían rendir las pruebas tradicionales para cumplir exigencias de regularidad de la asignatura.

3. BREVE DESCRIPCIÓN DE *ColoquiosEC*

El software permite realizar un seguimiento de los alumnos sobre el aprendizaje de temas abordados en Elementos de Computación mediante evaluaciones rápidas llamadas coloquios. Cada alumno rinde cuatro coloquios cuyos resultados deben ser informados a la cátedra.

El programa presenta una descripción detallada explicando su uso, a través de 5 pantallas de navegación secuencial. La última pantalla presenta un listado de todas las herramientas disponibles acompañada por una descripción sucinta, como se muestra en la Figura 1.

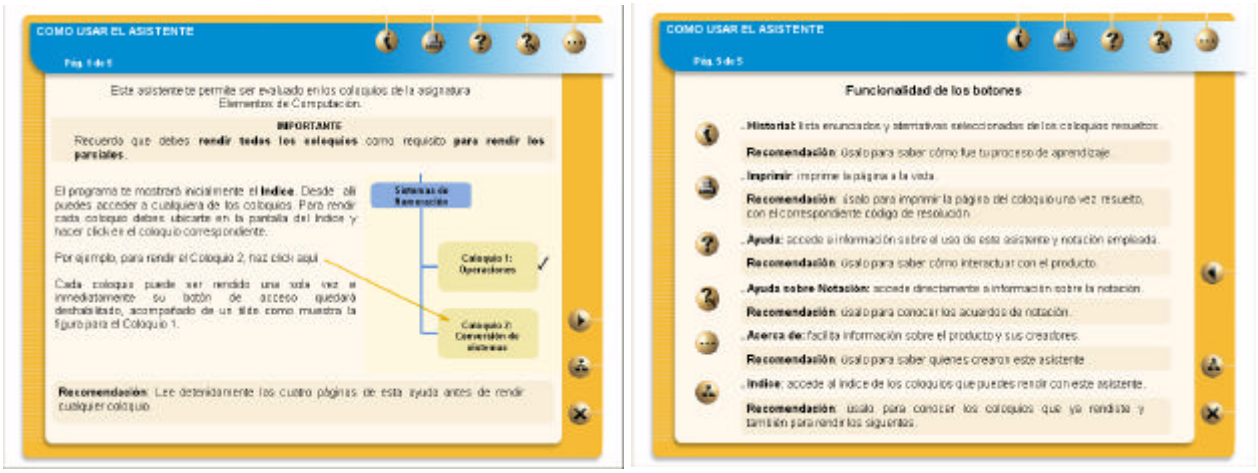


Figura 1: Pantallas del software *ColoquiosEC*

Para acceder a los coloquios el alumno dispone de un índice. Una vez elegido el coloquio se presenta una pantalla en la que se solicita la clave de acceso. Esta clave tiene el propósito de habilitar el acceso al coloquio en el momento definido por la cátedra. Los profesores son los encargados de divulgar dicha clave en el momento oportuno, a partir del cual los alumnos disponen de 48 horas para realizar la evaluación y devolver el resultado obtenido.

Cada estudiante tiene una única oportunidad para resolver el coloquio seleccionado. La respuesta del ejercicio debe ser elegida cuidadosamente, ya que una vez marcada no cuenta con opción de cambio e inmediatamente el programa evalúa la solución propuesta y devuelve un mensaje indicando si aprobó o no el coloquio y un código de resolución. El coloquio resuelto queda marcado y el alumno no puede rendirlo nuevamente.

ColoquiosEC se distribuye en CD. La cátedra lleva registro de la correspondencia biunívoca entre alumno y CD. El código de resolución que genera el programa depende del número de identificación único del CD, del número de problema que el software elige al azar de entre una colección y de la respuesta seleccionada por el alumno. Mediante este código, la cátedra dispone de

un programa que le permite identificar el alumno evaluado, el ejercicio que resolvió y la respuesta seleccionada.

Una vez resuelto el ejercicio, el alumno debe acercar la clave de resolución a la cátedra, para lo cual cuenta con la posibilidad de realizar una impresión de la pantalla o copiar esta clave en forma manuscrita y presentarla al docente responsable de su recepción en forma personal o enviando un mensaje por correo electrónico a la cuenta de la cátedra.

En las siguientes imágenes se puede apreciar la secuencia de pantallas que el producto presenta al alumno durante el proceso de evaluación:

1. Esta imagen muestra el índice desde el cual se selecciona el coloquio.

2. El coloquio se habilita con la introducción de la clave de acceso.

3. Pantalla de resolución en la que el alumno elige una alternativa de respuesta.

4. Devolución de la evaluación, código de resolución e indicaciones para la entrega del mismo.

Es importante destacar que este software no necesita ser instalado, ya que se ejecuta desde el CD. Esta característica facilita su uso ya que le permite independizarse de la versión de sistema operativo que posea la computadora y también libera al alumno para que acceda desde diferentes equipos, personal, de la universidad, cyber, etc.

4. LA EVALUACIÓN INTEGRAL

Retomando nuestra introducción, en la que reconocimos diferentes intereses por la evaluación, respecto a su naturaleza, planteamos las siguientes líneas de investigación:

- Evaluación del aprendizaje del alumno
- Evaluación de la acción formativa
- Evaluación del material educativo

Los resultados que se brindan a continuación provienen de dos fuentes: una encuesta aplicada al alumnado sólo en el año 2006, cuya finalidad fue la de indagar sobre la valoración del producto *ColoquiosEC* y la información sobre el rendimiento registrado en la base de datos de la cátedra en ambos años, tanto de los coloquios tradicionales como de los evaluados virtualmente.

Alumnos encuestados: 200. De esta muestra, 30 pertenecen al grupo piloto de los 50 alumnos seleccionados para adoptar la nueva modalidad.

4.1 Evaluación del aprendizaje del alumno

La aprobación de un mínimo de 50% de los coloquios impartidos, es condición necesaria para que el alumno resulte habilitado a rendir el parcial de la materia que incluye entre sus contenidos los evaluados en dichos coloquios. Por lo tanto, una porción del estudiantado no accede a la evaluación del parcial y consecuentemente resulta libre. Previo al parcial están previstas instancias de recuperación de coloquios. Los rendimientos observados en el año 2006 y 2007 se muestran en los Gráficos 1 y 2 respectivamente.

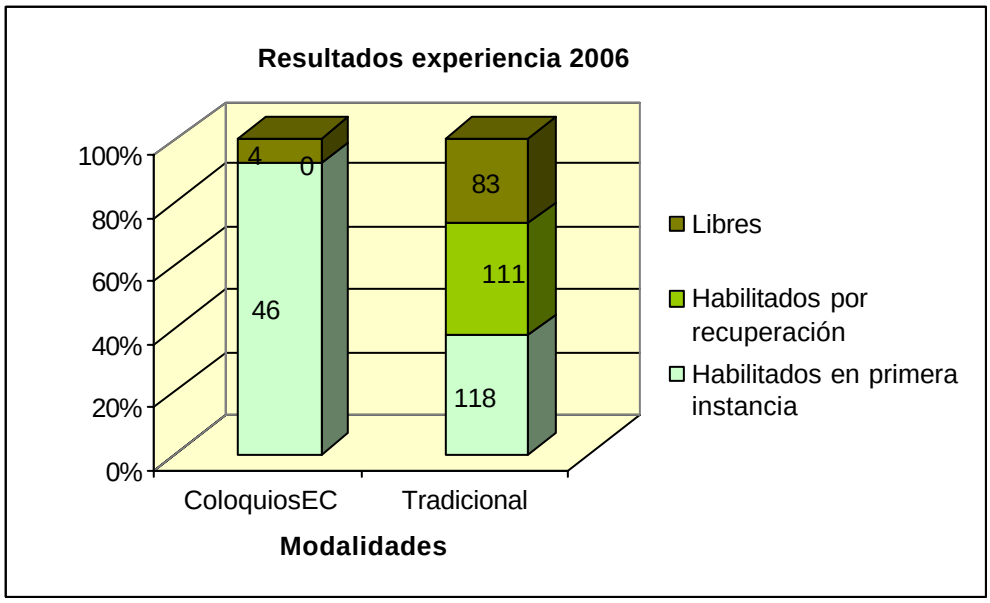


Gráfico 1. Resultados Experiencia 2006

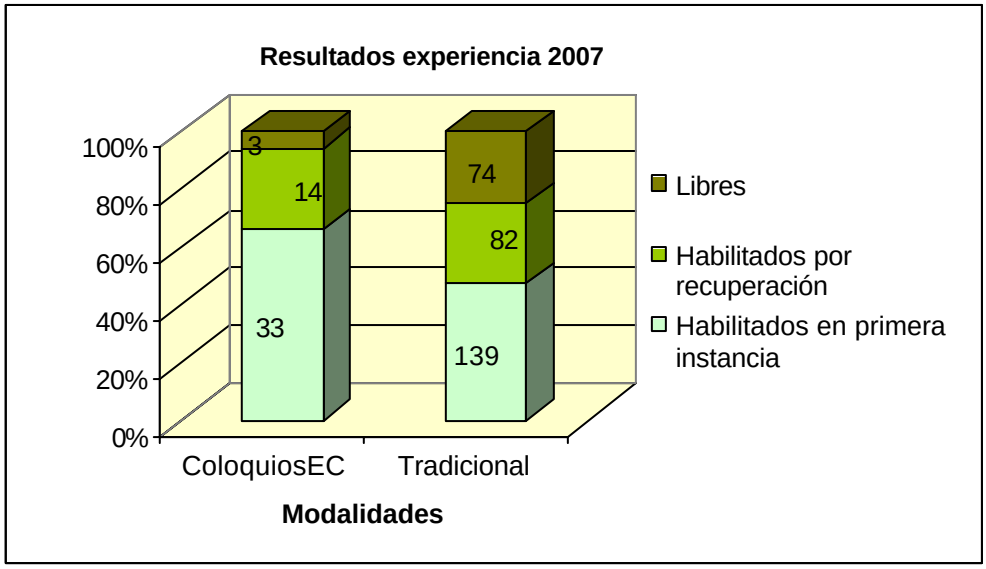


Gráfico 2. Resultados Experiencia 2007

Cabe destacar que el programa *ColoquiosEC* no brinda al alumno la posibilidad de recuperar el coloquio rendido. Esta decisión de diseño se sustenta en la intención de modificar drásticamente el objetivo de estas evaluaciones cortas y la percepción que el alumno tiene de ellas. Hasta el presente, si bien los coloquios se incorporaron para contribuir con el seguimiento de los aprendizajes, su impacto sobre la condición del estudiante en la materia prevalece, actuando más como un filtro para los alumnos rezagados que como un mecanismo de ayuda a los aprendizajes. Por lo tanto, abandonar la modalidad tradicional de evaluación y reemplazarla por una evaluación virtual pone el acento en nuestros principales objetivos, es decir, propiciar el trabajo autónomo, favorecer el entorno de la instancia evaluativa y fortalecer el seguimiento exigiendo la participación y no la aprobación. Los resultados obtenidos apoyan estos principios. Como puede verse en el Gráfico 1, 46 de 50 alumnos, es decir el 92%, tuvieron un excelente rendimiento usando *ColoquiosEC* en

comparación al 38% de los aprobados en primera instancia, para la modalidad tradicional, en el año 2006. Puede observarse que en el Gráfico 2 que en el año 2007 se obtuvo un buen rendimiento, aunque menor al del año anterior, con un 66% de alumnos que accedieron al primer parcial en la primera instancia contra un 47% correspondiente en modalidad presencial. Comparativamente, se puede notar una disminución en el rendimiento del grupo 2007 en lo que respecta al uso del producto, sin embargo, el número de estudiantes habilitados por primera instancia y recuperación de coloquios, para rendir el primer parcial, es levemente superior en el último año. Esta diferencia puede surgir de varios factores, entre los cuales se deben tener presente la realidad socio-cultural y económica de cada grupo, el efecto de haber aplicado un producto que en su segundo año ya no novedoso y una actitud tal vez menos comprometida por parte de los estudiantes en lo que respecta a su formación académica.

El impacto del uso de las NTICs en procesos evaluativos también fue motivo de investigación. El grupo piloto, como ya se dijo, sólo en esta oportunidad de experimentación tenía una doble exigencia, rendir virtual y tradicionalmente cada coloquio. De la comparación de rendimiento entre ambas modalidades surgen los siguientes valores:

Comparación	Rendimiento		
	Menor	Igual	Mayor
Presencial vs Virtual (Año 2006)	16/50 32%	14/50 28%	20/50 40%
Presencial vs Virtual (Año 2007)	3/50 6%	12/50 24%	41/50 82%

Tabla 1. Comparación de rendimiento entre modalidades

Puede notarse que en el primer año (2006) la acción mediadora del material no influyó significativamente en el rendimiento. Podría deducirse que en el segundo año el uso del producto tuvo un impacto positivo en el rendimiento de los coloquios tradicionales, sin embargo queda en duda la fuerza de tal impacto debido a que, como se explicará mas adelante, algunos alumnos no presentaron a la cátedra los resultados obtenidos con el uso del material.

4.2 Evaluación de la acción formativa

En este apartado centraremos el análisis de los mecanismos de comunicación entre docentes y alumnos en el marco de esta experiencia, habida cuenta de que las NTICs suponen la comunicación con una dinámica distinta.

ColoquiosEC habilita la posibilidad de enviar por correo electrónico los resultados de las evaluaciones rendidas. Durante la experiencia del año 2006, los alumnos podían optar por esta vía o entregar los resultados en forma personal. Sólo en nueve oportunidades, los alumnos adoptaron el servicio de e-mail para tal fin. En la inmensa mayoría de los casos hicieron llegar sus resultados en forma personal. Como conclusión de esta experiencia en el primer año, se planteó el interrogante sobre si la baja tasa de envíos por correo electrónico obedece más a limitaciones de acceso a Internet o a una decisión de adoptar o no esta vía de comunicación por cuestiones culturales. Si la respuesta es lo primero, nuestro desafío es investigar en profundidad antes de implementar un sistema de evaluación basado completamente en los servicios de la red, mientras que si es lo segundo cabe diseñar estrategias de fortalecimiento de comunicación entre docentes y alumnos y entre pares apropiadas a la modalidad.

Año 2006		
Coloquio Nº	Cantidad de presentaciones fuera de plazo	Cantidad de presentaciones no efectuadas
1	9/40 = 22,5%	10
2	0/30 = 0%	20
3	5/22 = 22,7%	28
4	2/21 = 9,5%	29

Tabla 2. Respuesta a las exigencias de presentación, año 2006

Durante el año 2007 y en pos de esclarecer el interrogante mencionado anteriormente, se restringió a los alumnos a presentar los resultados sólo mediante el uso de e-mail, para lo cual se habilitó un laboratorio con los recursos necesarios.

Año 2007		
Coloquio Nº	Cantidad de presentaciones fuera de plazo	Cantidad de presentaciones no efectuadas
1	15/31 = 48,38%	19
2	8/23 = 34.78 %	27
3	5/15 = 33,33 %	35
4	1/6 = 16, 67%	44

Tabla 3. Respuesta a las exigencias de presentación, año 2007

Los resultados registrados muestran que la falta de acceso a la tecnología no es la primera causa de incumplimiento en las presentaciones, sino mas bien, cuestiones culturales y actitudinales llevan a que los alumnos no se comprometan con las actividades que le fueron requeridas desde la cátedra, tales como presentación de resultados en fechas preestablecidas.

Como puede observarse, la adecuación al sistema fue paulatina. Los alumnos están acostumbrados a ser evaluados por resultados y no por procesos, por lo que aquellos que alcanzaban el mínimo de 50% de coloquios aprobados (2 de 4) en la modalidad presencial, no continuaba presentando los resultados de los coloquios evaluados por el software. No tenemos evidencias de si tales evaluaciones fueron o no rendidas, a lo sumo podemos ver los valores de la tabla que hablan de un compromiso decreciente con el sistema de seguimiento.

Un resultado sumamente significativo corresponde al comportamiento de los alumnos no pertenecientes al grupo piloto. De la encuesta tomada en el año 2006 surge que 75 de 170 alumnos, un 44%, utilizó *ColoquiosEC* como instrumento de auto evaluación. Este dato se complementa con el que surge de medir la cantidad de ellos que informaron a la cátedra los resultados de sus pruebas. Sólo para ilustrar, un único alumno presentó el resultado de tercer y cuarto coloquio. Este fenómeno, no sólo ratifica nuestra afirmación de que los alumnos no ven a la evaluación centrada en el proceso sino que además indica su gran interés en materiales que les provean actividades evaluativas.

Desde una perspectiva didáctica, creemos que la incorporación de las NTICs en instancias de evaluación obligan a una acción formativa que incluya estrategias de fortalecimiento del aprendizaje autónomo y sistemático, más propio de los estudios superiores que los presentes en el

alumno universitario inicial.

4.3 Evaluación del material educativo

Finalmente, nuestra experiencia permitió detectar fortalezas y debilidades del producto desarrollado que sirven de insumos para su mantenimiento perfectivo. Brindaremos una mirada cuantitativa sobre las valoraciones realizas por los alumnos encuestados y presentaremos las sugerencias de mejoras mas observadas.

En una escala de 1 a 5, el programa *ColoquioEC* obtuvo un puntaje promedio de 4,16 dentro del grupo piloto y 4,27 fuera de él. Esta semejanza es consistente con el alto valor que los alumnos dan al software, independiente del sistema de presentación exigido para los coloquios virtuales.

Respecto a la facilidad de uso del producto, principalmente lo amigable de su interfaz, 24 de 30 alumnos encuestados en el grupo piloto se manifestaron por su buena calidad. Comparativamente con ese 80%, el 61% del resto de los alumnos lo hizo en el mismo sentido.

Consultados sobre las bondades del material, las respuestas mas observadas son:

1. *Provee práctica*
2. *Sirve para auto evaluarse*
3. *Facilita llevar la materia al día*
4. *Se puede rendir en casa*
5. *Se rinde sin presiones*
6. *Provee de ejercicios que ayudan a preparar el parcial*

Respecto a las cualidades negativas que encuentran en el producto, señalan:

1. *Provee pocos ejercicios*
2. *No brinda la solución del ejercicio ante una respuesta incorrecta*
3. *No permite recuperación del coloquio ante una respuesta incorrecta*

Sobre la primera cualidad negativa, entendemos que, dado que la mayoría del alumnado no estaba obligado a presentar sus resultados, usaron *ColoquiosEC* como software ejercitativo, de allí que naturalmente la cantidad de ejercicios les resultara escasa. Análogamente, la segunda de las cualidades condice con el uso que los alumnos dan al software, quienes lo ven como ejercitativo y pretenden encontrar una devolución detallada al finalizar la práctica. La última se enmarca una vez más, en uso del software por alumnos que entienden las instancias evaluativas orientadas al producto más que al proceso.

Las sugerencias relevadas reiteran la necesidad manifiesta de los estudiantes de contar con una mayor cantidad de actividades prácticas. El alumno que inicia sus estudios de futuro analista de sistemas no tiene aún una visión clara acerca de la importancia de construir una estructura de pensamiento lógico-formal. Aplica su razonamiento intentando abarcar exhaustivamente todos los casos posibles de un problema sin construir conceptos que desde la lógica le permitan generalizar. Esta conducta, cimentada en el aprendizaje de la matemática como un esfuerzo de reforzamiento lo lleva a sobreestimar la cantidad de ejercitación por encima de la calidad de la misma, posición contrapuesta con la metodología de la cátedra que naturalmente subyace en el diseño del software.

A modo de conclusión, creemos que esta experiencia se puede aplicar a todo el alumnado. Teniendo presente que se trata de un ingreso masivo, nos animamos a reemplazar el "se puede" por el "se debe". Las estadísticas de rendimiento del grupo piloto arrojan un 60% de alumnos aprobados en el parcial en el que se evalúan los contenidos monitoreados por *ColoquiosEC*. Comparativamente, el resto de los alumnos pasaron esta instancia con éxito en el 65% de los casos, confirmando nuestra hipótesis que la incorporación del software no obstaculizó el acto de aprender. Podría pensarse que tampoco lo favoreció significativamente, lo cual nos llevaría a preguntarnos ¿dónde está el beneficio de este cambio?. La respuesta creemos, pasa por aprovechar los nuevos mecanismos de comunicación para zanjar las distancias de las que habla Fainholc. Estamos concientes del riesgo de imprimirle mayor anonimato a un alumno que ya hoy se enfrenta con las dificultades de la presencialidad en condiciones adversas. Este alumno, que asiste a clases en las que cumple un rol totalmente pasivo, podría optar por más y mejor participación si se le brindan los materiales adecuados y se los usa en el marco de una estrategia metodológica innovadora. Esta innovación debe venir de la mano de un cambio profundo en la percepción de la educación superior como formadora de profesionales "comprometidos" ante la sociedad, en una capacitación de calidad, conciente y continua. Para ello, se debe trabajar desde los primeros años universitarios, concientizando al alumno y preparándolo para que sea un actor activo y protagónico en su proceso de aprendizaje.

REFERENCIAS

[1] Briones, Stella y Martínez, M. Teresa. La educación a distancia: hacia un encuentro de sentidos. En III Seminario Internacional de Educación a Distancia, Universidad Nacional de Córdoba, Argentina, 26 y 27 de Mayo de 1998, pag.17.

[2] Fainholc, Beatriz. La Interactividad en la Educación a Distancia (en Litwin, Edith, Maggio, Mariana y Roig, Hebe (compiladoras). Educación a Distancia en los 90. Desarrollo, problemas y perspectivas) Buenos Aires: Facultad de Filosofía y Letras de la Universidad de Buenos Aires. Programa UBA XXI.1994, pag.131.

[3] Litwin, Edith. La Educación a Distancia. Temas para el debate en una nueva agenda educativa, Ed. Amorrortu, Buenos Aires, 2000.

[4] Litwin, Edith. Diseño e implementación de propuestas en línea de educación a distancia. www.educ.ar

[5] Litwin, Edith. Las nuevas tecnologías y las prácticas de la enseñanza en la universidad. www.unsavirtual.org

[6] Mac Gaul, Marcia; López, Marcela; Fernández, Eduardo. Las Nuevas Tecnologías al servicio del alumno ingresante. De la modalidad presencial a la modalidad a distancia... ¿o viceversa?. Congreso Nacional de Articulación de la Educación Superior. Salta. 2004.

[7] Mena, Marta. La convivencia institucional de las modalidades de educación presencial y a distancia: competencia o cooperación? (en Litwin, Edith, Maggio, Mariana y Roig, Hebe (compiladoras) Educación a Distancia en los 90. Desarrollo, problemas y perspectivas) Buenos Aires: Facultad de Filosofía y Letras de la Universidad de Buenos Aires. Programa UBA XXI. 1994.

[8] Prieto Castillo, Daniel. La comunicación en la educación. Editorial Ciccus, La Crujía, Buenos Aires, 1999.

[9] Revista Zona Educativa. Año 3 N° 21

[10] Watson, María Teresa. Historia de la educación a distancia en Argentina: un contexto de surgimiento. Revista RUEDA N°6/2005.

Diseño de un entrenador

Zulema Beatriz Rosanigo
brosanigo@infovia.com.ar

Alicia Beatriz Paur
apaur@ar.inter.net

Pedro Bramati
pedrobramati@speedy.com.ar

Hernán Bramati
hernan.bramati@gmail.com

Facultad de Ingeniería – Sede Trelew – U.N.P.S.J.B. Te-Fax (02965) 42 84 02

Resumen

En este artículo se expone el diseño de un software educativo, ETG, entrenador y generador de tutoriales, producto resultante del proyecto de investigación PI Nº 530 "*Generador de entrenadores basados en componentes reusables*" de la Universidad Nacional de la Patagonia San Juan Bosco, Argentina.

ETG facilita la tarea de crear tutoriales como recurso didáctico, que colaboren y potencien los procesos de aprendizaje en la educación superior y sirvan de apoyo al proceso de enseñanza-aprendizaje. Al mismo tiempo, el alumno al interactuar con esos tutoriales, puede analizar el problema y la solución propuesta, tantas veces como lo desee, observando el comportamiento, incrementando así su capacidad de asimilación y fijación de conceptos. También permite que el alumno realice prácticas sobre el tema, y en caso de cometer errores, puede intervenir guiándolo hacia la solución correcta, sin necesidad de la permanente presencia del docente, a la vez que le brinda un mecanismo de auto evaluación.

Palabras claves: Tutorial interactivo – entrenamiento - framework – software educativo

1 Introducción

La innovación en los ambientes de enseñanza-aprendizaje con la utilización de nuevas tecnologías motoriza la investigación, análisis y evaluación de nuevas herramientas que permitan mejorar la formación de los futuros profesionales, y provoca la actualización de los profesores de las diferentes áreas curriculares con la finalidad de incorporar estos elementos a sus actividades docentes.

Por esta razón, nuestro grupo de investigación tiene como principal línea de investigación, la aplicación de las nuevas tecnologías en la educación. En esta dirección se han desarrollado varios proyectos en los que se evaluaron diferentes medios y se diseñaron contenidos y recursos didácticos a aplicar en la enseñanza superior, como apoyo a la tarea docente tanto en modalidad presencial como no presencial.

Siguiendo con la línea de investigación del proyecto previo: "Construcción de tutoriales basados en componentes reusables", cuya finalidad fue facilitar la construcción de herramientas educativas, en este proyecto se diseñó un framework integrado a TutGen (generador de tutoriales) para abarcar el entrenamiento controlado y asistido que el alumno necesita realizar para ejercitarse y adquirir la destreza pretendida.

Se trata de un entrenador genérico e inteligente, denominado ETG, que interpreta de cada tutorial los diferentes caminos que conducen a la solución, los reconoce como válidos sin necesidad de que el docente deba expresarlos uno por uno, y asiste al alumno en sus errores guiándolo hacia la solución correcta, sin necesidad de la permanente presencia del docente, a la vez que le brinda al alumno un mecanismo de autoevaluación.

El tipo de problemas que se abarca es de naturaleza procedimental y de dominios que implican la modelización de procesos físicos y conocimiento procedural en el cual, los aspectos de comportamiento, resultan cruciales. Se puede utilizar en aquellas áreas del conocimiento donde se presentan problemas cuya solución requiere seguir determinado procedimiento o proceso paso a paso o también para adquirir destrezas.

La dificultad de construir un entrenador reside en que el sistema debe tener conocimiento del dominio, así como un mecanismo de interacción con el estudiante que le permita transmitirlo y verificar que esta comunicación fue eficaz, y de no ser así, efectuar alguna acción correctiva. Porque se trata de un software educativo, además de los requisitos funcionales y operativos, es sumamente importante considerar los aspectos pedagógicos en etapas tempranas del ciclo de vida, ya que tendrán incidencia a lo largo de toda la vida del mismo.

En este artículo se describe el diseño del generador de tutoriales y entrenadores.

2 Fundamentos

2.1 Teorías de aprendizaje

El propósito de las teorías educativas es el de comprender e identificar los procesos de adquisición de conocimiento y a partir de ellos, tratar de describir métodos para que el aprendizaje sea más efectivo.

Nuestro deber como docentes es aprovechar de cada teoría lo que mejor convenga a la situación, al sujeto, y al tipo de aprendizaje a lograr, enseñando al alumno a pensar y actuar en forma autónoma a fin de que adquiera más conocimientos con creciente capacidad y destreza

Según Salcedo Lagos [23] las aproximaciones al fenómeno del aprendizaje oscilan entre dos extremos: del conductismo al cognitivismo y viceversa.

En un extremo, la teoría del conductismo [25], lo único que ve del sujeto que aprende son las condiciones externas que favorecen su aprendizaje. Habla de un modelo de "caja negra" donde lo fundamental es la programación en pequeños pasos que llevan al logro del objetivo esperado, lo que se manifiesta por la respuesta del sujeto y su reforzamiento.

En el otro, para la teoría del cognitivismo [14] lo que cuenta es el sujeto con todo su campo vital, su estructura cognitiva y las expectativas que tiene. Se habla de un modelo de "caja traslúcida" donde lo que cuenta es el sujeto dentro de su entorno psicológico y social.

En medio de los enfoques anteriores, es posible encontrar una combinación de ambos, donde no se ignora por completo al sujeto, pero tampoco se lo hace eje principal.

Las teorías más representativas del proceso de enseñanza-aprendizaje son:

- Conductismo (Skinner 1958)
- Aprendizaje significativo (Ausubel 1983)
- Cognoscitivismo (Gardner 1987)
- Constructivismo (Piaget 1989)

La distinción básica entre estas grandes corrientes radica en la forma en que se concibe el conocimiento. Para el conductismo, el conocimiento consiste fundamentalmente en una respuesta pasiva y automática a factores o estímulos que se encuentran en el ambiente. Ausubel plantea que el aprendizaje depende de la estructura cognitiva previa del alumno que se relaciona con la nueva información, entendiendo por "estructura cognitiva", al conjunto de conceptos e ideas que un individuo posee en un determinado campo del conocimiento, así como su organización. El cognoscitivismo, por su parte, considera el conocimiento básicamente como representaciones simbólicas en la mente de los individuos. Finalmente, el constructivismo, como el término lo sugiere, concibe al conocimiento como algo que se construye, algo que cada individuo elabora a través de un proceso de aprendizaje.

2.2 Estilos de aprendizaje

Los estilos de aprendizaje son una combinación de características cognoscitivas, afectivas y una conducta psicológica que sirven como indicadores relativamente estables sobre cómo los aprendices perciben, interactúan y responden a su ambiente [1]. Nuestra herencia, experiencias de vida y las demandas del ambiente determinan en parte cómo percibimos y procesamos la información. Los estilos de aprendizaje se ajustan dependiendo de la tarea a realizar y al estilo de enseñanza que se utiliza.

Se ha descubierto que las personas se concentran más en una determinada etapa del proceso de aprendizaje, y en función de la etapa dominante se puede hablar de cuatro estilos de aprendizajes:

- Vivir la experiencia → ESTILO ACTIVO: Son de mente abierta, nada escépticos y acometen con entusiasmo nuevas tareas. Crecen con los desafíos y se aburren con los plazos largos. Les gusta trabajar en grupo, se involucran en los asuntos de los demás y centran a su alrededor todas las actividades. Aprenden mejor cuando la actividad les presenta un desafío, con actividades cortas de resultado inmediato. Les cuesta más trabajo aprender cuando tienen que adoptar un papel pasivo, cuando tienen que asimilar, analizar e interpretar datos o cuando tienen que trabajar solos. La pregunta que quieren responder con el aprendizaje es **¿Cómo?**
- Reflexión → ESTILO REFLEXIVO: Son prudentes y consideran todas las alternativas antes de dar un paso. Recogen datos y los analizan antes de llegar a una conclusión. Aprenden mejor cuando pueden adoptar la postura de observador u ofrecer observaciones y analizar la situación

y cuando pueden pensar antes de actuar. Les cuesta más aprender cuando son el centro de la atención y cuando se les apresura de una actividad a otra o tienen que actuar sin poder planificar previamente. La pregunta que quieren responder con el aprendizaje es **¿Por qué**?

- Generalización, elaboración de hipótesis → ESTILO TEÓRICO: Adaptan e integran las observaciones (hechos) dentro de teorías coherentes. Enfocan los problemas de forma vertical escalonada, por etapas lógicas. Son perfeccionistas y les gusta analizar y sintetizar. Aprenden mejor a partir de modelos, teorías, sistemas, con ideas y conceptos que presenten un desafío y cuando tienen oportunidad de preguntar e indagar. Les cuesta más aprender con actividades que impliquen ambigüedad e incertidumbre. La pregunta que quieren responder con el aprendizaje es **¿Qué?**
- Aplicación → ESTILO PRAGMÁTICO: Su punto fuerte es la aplicación práctica de las ideas. Descubren el aspecto positivo de las nuevas ideas y aprovechan la primera oportunidad para experimentarlas. Aprenden mejor con actividades que relacionen la teoría y la práctica, viendo a los demás hacer algo y cuando tienen la posibilidad de poner en práctica inmediatamente lo que han aprendido. Les cuesta más aprender cuando lo que aprenden no se relaciona con sus necesidades inmediatas o no tienen una finalidad aparente o no está relacionado con la 'realidad'. La pregunta que quieren responder con el aprendizaje es **¿Qué pasaría si...?**

Según Kolb [17], ningún modo de aprender es mejor que otro y la clave para un aprendizaje efectivo es ser competente en cada modo cuando se requiera.

Una buena experiencia de aprendizaje reta las capacidades del aprendiz, por lo tanto, se debe establecer un balance al propiciar maneras alternas para aprender y así lograr que los estudiantes ganen confianza, muestren interés en aprender, desarrollen destrezas de razonamiento, análisis, solución de problemas y exploren otros estilos de aprendizajes.

El conocer sobre los estilos de aprendizaje nos permite facilitar el proceso de enseñanza - aprendizaje y nos ayuda a reflexionar sobre nuestras funciones y responsabilidades como educadores ante las necesidades e intereses de los estudiantes. Para mejorar el aprovechamiento académico, aumentar el interés, participación y nivel de satisfacción en el alumno, los estilos de enseñanza del profesor y los estilos de aprendizaje de los estudiantes deben corresponderse y complementarse. Por lo tanto, debemos ser conscientes sobre las diferencias entre los estilos de aprendizaje y las estrategias de enseñanza que tenemos disponibles. El balance entre ambos aspectos estimula la colaboración y la participación del aprendiz en el proceso de enseñanza - aprendizaje, haciendo que sea más significativo y efectivo.

Teniendo en cuenta que un aprendizaje óptimo, según Kolb, requiere de las cuatro fases, debemos presentar el material a aprender de forma tal que garanticemos actividades que cubran las distintas fases de la rueda de Kolb.

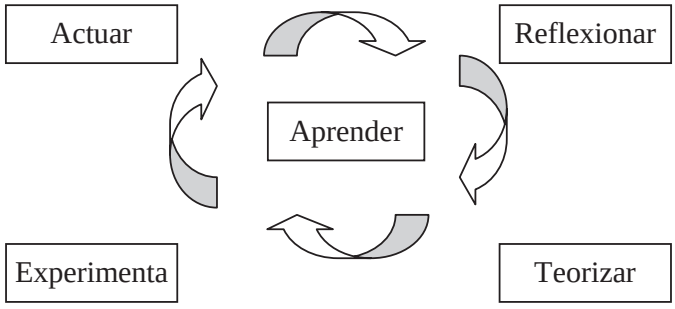


Figura 1 Fases del aprendizaje

De esta manera, facilitaremos el aprendizaje de todos los alumnos ya que habrá actividades en su estilo preferido y, además, les ayudaremos a potenciar las fases con los que se encuentran menos cómodos.

2.3 Enfoques para el diseño de material educativo

Existen diversos enfoques para el diseño de material educativo hipermedia [12, 21]:

- Basado en el diseño de los contenidos educativos: se articulan en cursos, lecciones, ejercicios y tests. El modelo de contenido está orientado de manera similar a la organización de las bases de datos y centrado en la idea de la estructuración del dominio educativo.
- Basado en el modelo hipertexto, en el que se modeliza un dominio educativo como una red de componentes de una granularidad determinada y donde las interacciones del alumno vienen dadas por las decisiones que este realiza durante la navegación por el material.
- Centrado en el estudiante y en sus necesidades, el diseño se realiza adaptándolo a los conocimientos previos del estudiante y a las interacciones potenciales de éste con el entorno. Hay un análisis previo de las interacciones con el entorno desde un punto de vista pedagógico y esto permite incorporar algunos nuevos paradigmas de aprendizaje en el sistema.
- Entornos integrados de enseñanza, basados en la creación integrada de políticas de acceso a servicios conocidos en el ámbito de la red: foros de debate, sistemas de conferencia electrónica, servicios para compartir archivos, aplicaciones de comunicación sincrónica. Se orientan fundamentalmente hacia el soporte de trabajo en grupo, generalmente para dar servicio de intercambio de material entre los alumnos.

2.4 Los Sistemas Tutores Inteligentes (STI)

Un STI es un sistema experto en una materia, diseñado con el fin de impartir conocimiento y que exhibe alguna forma de inteligencia para guiar al estudiante en el proceso de aprendizaje. Permite emular el proceso de enseñanza - aprendizaje, adaptando el tipo y el contenido de la enseñanza a las necesidades específicas del alumno, decidiendo cuándo introducir nuevos conceptos o repasar los anteriores si éstos no han sido asimilados. Estos sistemas tienen en cuenta los conocimientos a enseñar (contenido pedagógico), la forma de enseñarlo (estrategia pedagógica), así como la información relevante sobre el alumno que está siguiendo el tutorial.

Los STIs tienen la capacidad de asesorar al alumno y guiarlo en su proceso de aprendizaje, pero es fundamental la detección del conocimiento erróneo para que el comportamiento del STI sea el esperado. En dominios declarativos en los que sólo se diagnostica conocimiento teórico se suelen realizar pruebas de tipo test. En cambio, en dominios procedimentales, es necesario comprobar las habilidades del alumno al abordar problemas concretos. En este caso, el diagnóstico correcto de las acciones del alumno es totalmente dependiente del conocimiento que tenga el módulo correspondiente del STI sobre el entorno virtual y del grado de integración entre ambos.

Varios autores [8,9,11,16,24] coinciden en que un STI debe cumplir ciertos requisitos:

- Poseer conocimiento tanto sobre la manera como sobre los conceptos a enseñar.
- Ser adaptativo. Debe adaptar el nivel de enseñanza y el nivel de conocimiento a las necesidades y ritmo de estudio del alumno.
- Capacidad de reacción. Debe ser capaz de intervenir si el alumno responde de forma imprevista o su evolución de conocimiento entra en conflicto con los objetivos del tutor.
- Eficacia. Debe disponer de un abanico de estrategias didácticas suficiente y de un mecanismo de selección de dichas estrategias lo más acertado posible para conseguir que el alumno aprenda apropiadamente.

- Motivador. Debe animar al alumno mediante mensajes de aliento para que su evolución sea positiva y rápida.
- Evaluador. Debe evaluar los conceptos enseñados mediante ejercicios y cuestionarios para comprobar si el alumno está realmente entendido los conceptos que se le están presentando.

Según Kaplan [16] la arquitectura de un STI está compuesta por cuatro módulos interconectados:

- La interfaz, o Módulo de Diálogo que permite a los usuarios, alumnos y docentes interactuar con el sistema.
- El modelo instruccional o pedagógico, que se encarga de definir la estrategia para transmitir el conocimiento a los usuarios. Se encarga de adaptar el sistema a las características concretas del alumno, seleccionando el material didáctico y evaluando los conceptos aprendidos mediante ejercicios y cuestionarios.
- El modelo experto o del dominio, el cual versa sobre la materia o curso que se impartirá. Representa el dominio que se pretende enseñar al alumno. Este dominio debe estar organizado de forma que su enseñanza resulte sencilla y clara. También debe mantener relaciones entre conceptos y prerrequisitos de aprendizaje.
- El modelo del estudiante, que refleja cuánto conoce el estudiante sobre el dominio, así como las experiencias cognitivas y de aprendizaje que ha llevado, del cual puede obtenerse un diagnóstico. Este módulo es el que permitirá STI adaptarse a las necesidades del aprendiz.

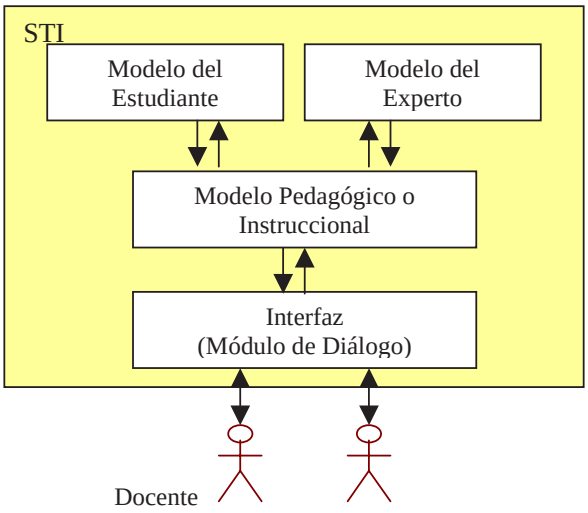


Figura 2 Arquitectura de un STI según Kaplan

Nuestro enfoque: Consideramos que el desarrollo efectuado se ubica en la categoría de software educativo interactivo e inteligente, o al menos con cierto grado de inteligencia, ya que tiene capacidad de aprender de las interacciones y pasos que el docente realiza para crear tutoriales y mostrar el proceso constructivo de algún tema, los recuerda y relaciona, y puede asistir al alumno en el entrenamiento, interviniendo y ayudando durante el proceso de ejercitación.

3 DISEÑO DEL ENTRENADOR

El diseño del sistema ha adoptado un enfoque orientado a objetos siguiendo la metodología de proceso unificado [4], aplicando patrones de software [5,6,13] y considerando los aspectos pedagógicos en etapas tempranas del ciclo de vida [12,15,18,22].

Los principales beneficios del modelo utilizado son:

- Incorporación y contemplación de aspectos pedagógicos, educativos y comunicacionales dentro de cada etapa del ciclo de vida.
- Incorporación de casos de uso, los que permiten reemplazar la especificación funcional tradicional, transformándose en guía para las actividades que se realizan durante el proceso de desarrollo, incluyendo diseño, implementación y pruebas del sistema.

- Estructuración de las actividades llevadas a cabo en el diseño y desarrollo del software, con la incorporación de componentes que permiten y privilegian el reuso de código.
- Importante disminución en el tiempo de desarrollo.
- Pronta incorporación de un prototipo que permite evaluaciones tempranas del producto y va evolucionando constantemente hasta convertirse en el producto deseado.

3.1 Arquitectura

La arquitectura del sistema es modular, de gran flexibilidad, con módulos independientes e interfaces bien definidas entre bloques, de forma tal que permite fácilmente cambiar los componentes que representan datos o transformaciones de datos de un dominio del conocimiento en particular. Se aplicó el patrón arquitectónico MVC (Model View Controller) [5], que divide el problema en tres componentes: el modelo que contiene el corazón de la funcionalidad, la vista que despliega la información al usuario y el controlador que maneja la entrada del usuario. El Modelo representa los datos o los objetos de la aplicación que son accedidos y presentados al usuario. La Vista es la representación en pantalla del estado actual del Modelo, y el Controlador es el objeto que manipula el Modelo y define la forma en que la interfaz de usuario reacciona ante las acciones del usuario. De esta manera se desacoplan los problemas y se logra mayor reusabilidad.

La vista y el controlador definen la interfaz gráfica, con la que tendrán que interactuar el docente, en el proceso de creación de un tutorial, y el alumno, en el proceso de aprendizaje, repaso o práctica. En la definición de la interfaz gráfica se tuvieron en cuenta las conclusiones arribadas al respecto en investigaciones previas.

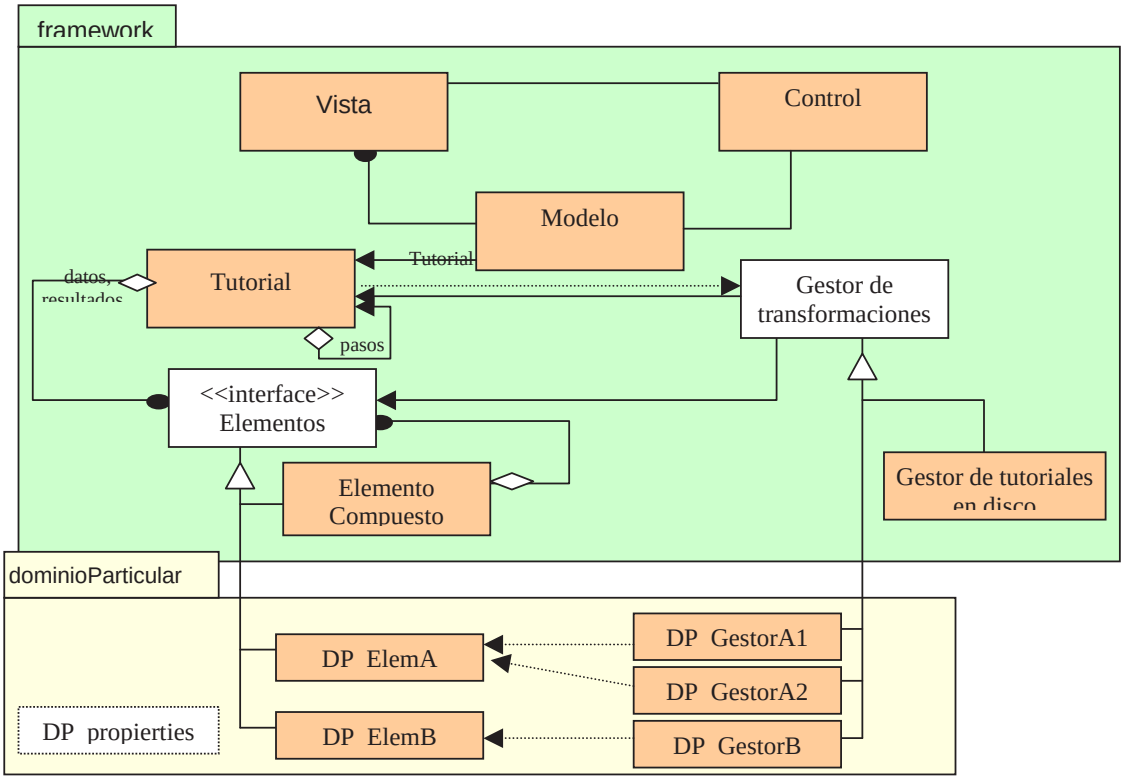


Figura 3 – Esquema de clases

Para permitir que pueda utilizarse en una variedad de dominios, su arquitectura tiene una clara separación entre las capacidades independientes del dominio y el conocimiento del dominio específico.

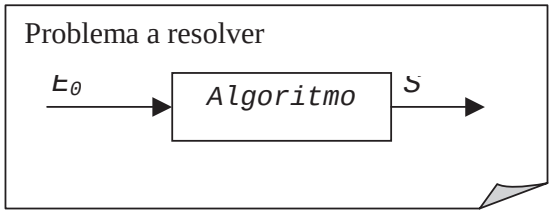
Se pueden distinguir módulos generales, comunes a todos los dominios de utilización de tutoriales y entrenadores, y módulos específicos que tienen en cuenta el comportamiento especial del dominio de aplicación. Los módulos generales describen un marco o "framework" y conforman el esqueleto de la aplicación, proveen la funcionalidad común, definen las abstracciones fundamentales y sus interfaces estableciendo las interacciones entre los objetos, dejando en determinados lugares espacios en blanco o puntos de articulación ("hot spot"). Cada espacio se refiere a los aspectos de los tutoriales y entrenadores que pueden variar de una aplicación a otra, y es allí donde la arquitectura debe proveer flexibilidad.

3.2 Características del Dominio de Conocimiento

Las tareas del dominio candidato o elegible deben ser descomponibles en subtareas y el campo de conocimiento debe ser estable, es decir, no deben existir demasiadas estructuras de conocimiento que se vuelvan obsoletas en el corto plazo o antes de que el sistema alcance el estado de utilización completa.

En nuestro caso, pretendemos que el sistema sea capaz de:

1. Generar procedimiento de resolución de un problema, a partir de un ejemplo que desarrolla el docente (experto del dominio), que comienza con la declaración de datos iniciales y la ejecución de sucesivas transformaciones hasta llegar al resultado esperado.



2. Ser capaz de reproducir el método con otros datos iniciales o como paso intermedio en la resolución de otro problema.
3. Guiar al alumno durante el aprendizaje del procedimiento.

Con esta herramienta se pretende enseñar a resolver un problema mediante un ejemplo, mostrando a partir de un conjunto de datos iniciales, la forma de llegar al resultado explicitando todos los pasos intermedios necesarios y enfatizando los conceptos relacionados.

Lo que se pretende mostrar son las transformaciones necesarias a realizar sobre E_0 para producir la salida S. Cada una de estas transformaciones es un paso del tutorial y habrá que realizarlas en un cierto orden, el cual puede no ser único.

Partiendo de esta base, es necesario identificar en el dominio los conceptos básicos que pueden intervenir como datos iniciales, y aquellos otros conceptos que representan transformaciones elementales sobre los primeros para producir nuevos conceptos, posiblemente más complejos.

3.3 Representación del conocimiento

Distinguimos de esta manera dos clases: *Elementos*, que representan los posibles datos de entrada y *Primitivas de Transformación*, que representa al algoritmo por el cual la entrada E_i del paso P_i se transforma en el elemento S_i.

El proceso completo de resolución de un problema queda encapsulado en la clase *Tutorial*, la cual tiene conocimiento de los Elementos que actúan como datos de partida y del conjunto de pasos que

conforman la resolución, así como el o los resultados que produce. Con colaboración de otras clases, el tutorial es capaz de almacenarse en disco y recuperarse para ser usado en otro contexto o como paso en la resolución de otro problema.

Mientras que la clase Tutorial es independiente de las características de un dominio, los Elementos y las Transformaciones tienen que especializarse en el dominio específico. Estas son las clases que aportan el comportamiento específico en un dominio particular.

Las interacciones de tutorización en un sistema de entrenamiento deben incluir explicaciones que describan el comportamiento de los componentes y los procesos físicos del sistema. Las explicaciones de comportamiento requieren la representación de modelos complejos o profundos en cooperación con los modelos conceptuales clásicos. Siguiendo esta idea, Vadillo Zorita et al. [26] proponen una representación del conocimiento del sistema formada por la coexistencia de múltiples modelos capaces de proporcionar diferentes puntos de vista del mismo con el objetivo de generar distintos tipos de explicaciones, identificando tres vistas diferentes del dominio:

- **Vista conceptual**: Agrupa todas las descripciones de los conceptos del sistema (componentes, procesos, procedimientos,...), y las relaciones entre ellos.
- **Vista estructural:** Indica cómo está organizado el sistema mediante la descripción de los componentes que lo constituyen y las conexiones físicas entre ellos.
- **Vista de comportamiento**: Agrupa las descripciones de comportamiento de las diferentes partes del sistema. Muestra cómo trabajan lo componentes en función de sus procesos asociados.

3.4 Conocimiento pedagógico

Los conceptos del dominio, tanto teóricos como de entrenamiento, además de estar representados adecuadamente deben organizarse de una manera pedagógica que ayude en el proceso de enseñanza de los mismos. Es necesario tener en cuenta aspectos relativos al orden en que deben ser presentados al alumno y las relaciones de comparación entre ellos, distinguiendo si un concepto es prerrequisito de otro o no, si es equivalente o no, si es más abstracto o concreto que otro, etc.

En nuestro sistema, algunas de estas relaciones son deducibles por el sistema en el proceso de generación del tutorial, pero muchas otras, requieren que los componentes propios del dominio, implementados como Elementos o Transformaciones, las establezcan, y así poder inferir, por ejemplo la equivalencia o no de dos procedimientos compuestos.

3.5 Adquisición de conocimiento

Los Elementos y Primitivas de Transformación aportan el conocimiento básico del dominio. La adquisición de más conocimiento por parte del sistema, se realiza durante el proceso de resolución de un problema mediante un ejemplo por parte del docente (experto). Mientras éste va seleccionando los datos y los pasos necesarios para su solución, el sistema va registrando estas acciones, tiene en cuenta precedencias y precondiciones y va generando un tutorial. Finalmente, cuando el docente indica que ha finalizado y desea guardarlo, el sistema genera dos archivos: un archivo XML y otro HTML. El primero, es utilizado por el propio sistema cuando se requiere utilizar el tutorial creado, como paso de solución de otro problema, o cuando el alumno requiere hacer prácticas de resolución en esa temática. Este archivo, tiene la información necesaria para reproducir la solución del problema o resolver otro problema del mismo tipo, como así también, reconocer si la solución que está planteando el alumno es la correcta o equivalente, y en caso contrario intervenir con algún tipo de ayuda o refuerzo. El segundo archivo, es la explicación del tutorial creado, que se utiliza cuando ejecuta el tutorial para mostrar o enseñar los pasos de

resolución, y que también puede ser utilizado como Objeto de Aprendizaje para ser compartido y reutilizado en otro contexto.

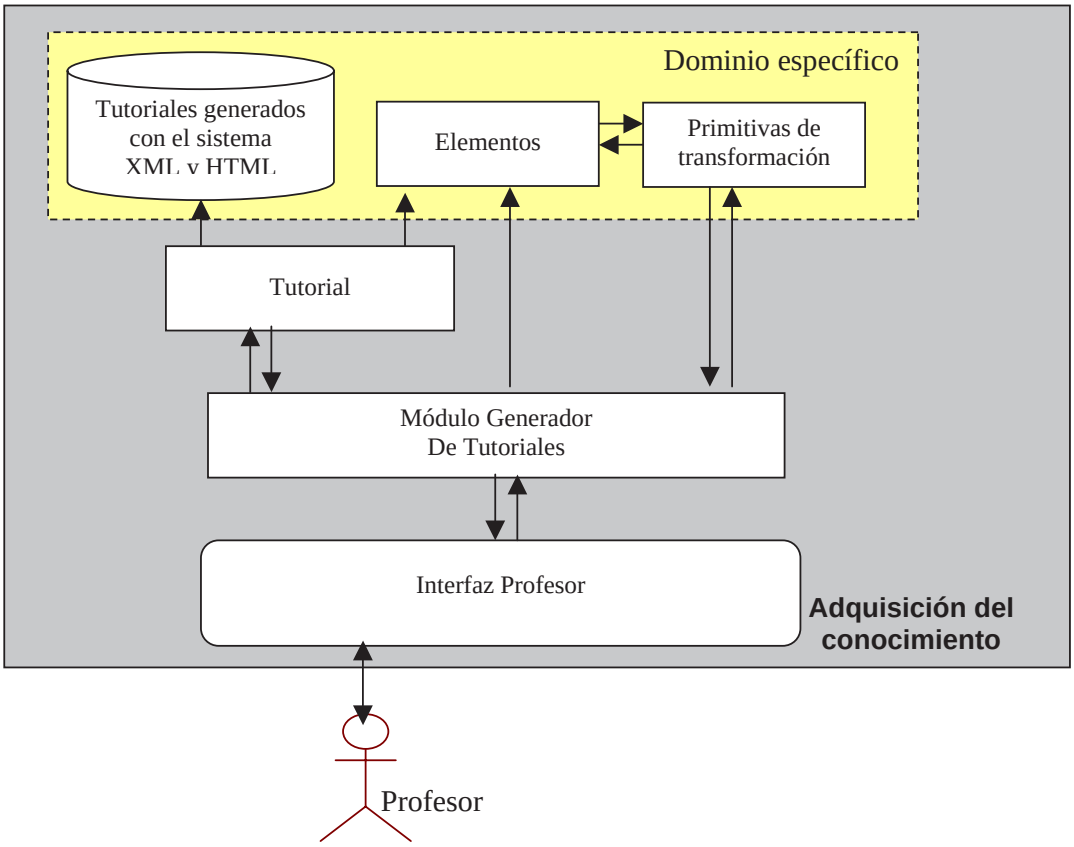


Figura 4 – Adquisición del conocimiento

3.6 Entrenamiento

Para el entrenamiento, el núcleo principal es el Tutor quien decide y guía el proceso de enseñanza y adiestramiento de cada alumno en particular. Este proceso de enseñanza y adiestramiento requiere una estrecha interacción entre el Tutor y el alumno en pos de conseguir los objetivos planteados. En este sentido las explicaciones dadas por el tutor son de una gran importancia debido a que su adecuación marcará el éxito en la consecución del objetivo primordial, que el alumno aprenda.

Cuando el alumno decide realizar una práctica, el tutor le ofrece elegir entre las disponibles para su perfil. Una vez elegida, el Tutor genera un tutorial con la información recuperada de la base XML, y lo tiene como procedimiento base contra el cual comparar las acciones del alumno como así también para mostrar la solución del problema en forma total o parcial, si el alumno lo solicitara. A medida que el alumno interacciona con el sistema para realizar los pasos u operaciones sobre el procedimiento en que está entrenando, el módulo Generador de Tutoriales, va registrando todas esas operaciones, de igual manera que lo hacía con el docente y creando un nuevo tutorial con los datos y pasos que el alumno selecciona. Por su parte, el Tutor está continuamente observando y por cada paso realizado por el alumno, compara con el tutorial base, analizando si es correcto o es factible de serlo, o si es un paso innecesario o si es definitivamente incorrecto. En función de ello, y la configuración de errores establecida, decide o no intervenir para ayudar y guiarlo en su aprendizaje.

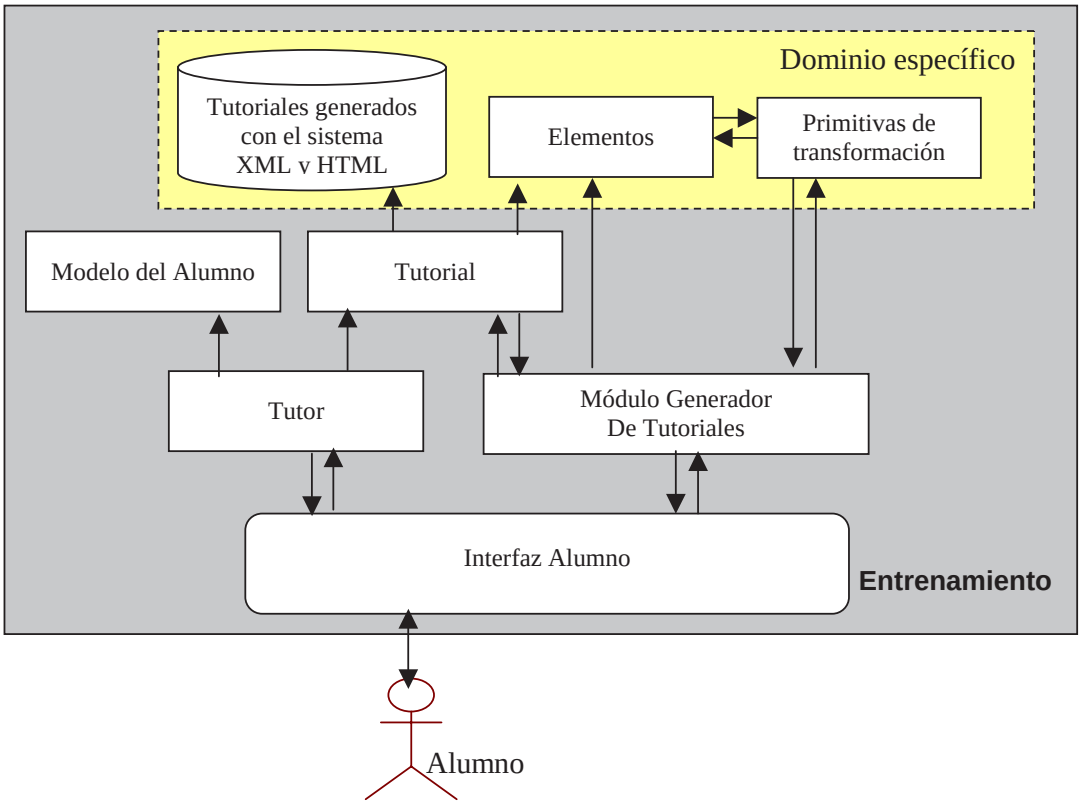


Figura 5– Entrenamiento

4 Conclusiones

El uso de tutoriales y entrenadores presenta importantes beneficios educativos ya que estimula y ayuda a los alumnos en las distintas situaciones del aprendizaje.

En nuestro desarrollo, el entrenador siempre puede interpretar los diferentes caminos que surgen de alterar el orden de las operaciones requeridas, teniendo en cuenta aquellas sub-secuencias que no podrían ser alteradas sin producir error, y también puede distinguir algunas formas equivalentes de realizar una tarea. De todas maneras, se encuentra restringido al mundo acotado que determina la inteligencia dada a los componentes específicos, ya que no sería capaz de inferir otras soluciones al mismo problema si no fueron previstas por el desarrollador del componente.

Más allá de las facilidades provistas por herramientas con características similares a ésta, el docente debe mantener su rol de mediador en el proceso de enseñanza-aprendizaje, ya que ninguna herramienta supliría su función.

5 Bibliografía

[1] Alonso C, Gallego D., Honey P. (1999). *Los estilos de aprendizaje: procedimientos de diagnóstico y mejora*, , Ediciones Mensajero, Bilbao.

[2] Ausubel-Novak-Hanesian (1983.) *Psicología Educativa: Un punto de vista cognoscitivo* .2° EdicionesTRILLAS México

[3] Booch G., Jacobson I., Rumbaugh J. (1998). *The Unified Modeling Language*, Addison-Wesley Publications

[4] Booch G., Jacobson I., Rumbaugh J. (1998). *The Unified Process Sofwtare Development* Addison-Wesley Publications,

[5] Buschmann F., Meunier R., Rohnert H., Sommerland, P., Stal, M. (1996) *Pattern-Oriented Spftware Architecture: a system of patterns.* Ed. Wiley

[6] Cooper, James W. (1998) - *Java Design Patterns: A Tutorial* – Addison Wesley

[7] Cruz Feliú, J. (1997) *Teorías del aprendizaje y tecnología de la enseñanza*. Trillas. México.

[8] Cuevas Vallejo, Carlos Armando (1996) "*Sistemas Tutoriales Inteligentes*" INVESTAV, I.P.N., consultado en www.matedu.cinvestav.mx/~ccuevas/STI96.htm

[9] Cumming G. D. y Self J. A. (1991), "*Learner models in collaborative intelligent educational systems*", http://citeseer.ist.psu.edu/cumming91learner.html

[10] Felder R. M. & Henriquez E. R. (1995) *Learning and Teaching Styles* in Foreign and Second Language Education, Foreign Language Annals, 28, No. 1.

[11] Fred, A. (1994) "*Expert Systems: Conceptual, Methodological and Management Aspects,*" en Project HC 1014 – ENN – European Neurological Network.

[12] Galvis, Alvaro (1994) *Ingeniería de Software Educativo*. Ediciones Uniandes.

[13] Gamma, Eric; Helm, Richard; Johnson, Ralph and Vlissides, John (1995) *Design Patterns. Elements of Reusable Software*, Addison-Wesley.

[14] Gardner H. (1987) "*La nueva ciencia de la mente: Historia de la psicología cognitiva*". Barcelona.

[15] Gómez, R., Galvis, A. y Mariño, O. (1999) Ingeniería de software educativo con modelaje OO. Brasil. Consultado en: http://www.minerva.uevora.pt/simposio/comunicacoes/rigomezmarino.html

[16] Kaplan Randy, Rock Denny (1995): "*New Directions for Intelligent Tutoring*". AI Expert, Febrero.

[17] Kolb D. (1984) *Experiential Learning: Experience as the Source of Learning and Development*, cit. por Felder Richard & Henriquez Eunice (1995)

[18] Marqués, Pere (1999) – *Programas Didácticos: Diseño y Evaluación* –Consultado en http://www.xtec.es/~pmarques/edusoft.htm

[19] Novak J. D., y Gowin D. B., *Aprendiendo a aprender* , 1988 Ed. Martínez Roca. Barcelona.

[20] Pressman, Roger S.*(1998), Ingeniería de software: Un enfoque práctico*. Ed. Mc Graw Hill.

[21] Rodríguez Artacho Miguel (2000) Una arquitectura cognitiva para el diseño de entornos telemáticos de enseñanza y aprendizaje - Tesis doctoral Universidad Nacional de Educación a Distancia - Escuela Técnica Superior de Ingenieros Industriales.

[22] Rosanigo, Z.B., Paur, A., Bramati, P. (2000) *Metodología de desarrollo de software educativo*. Actas de VI Congreso Internacional de Ingeniería Informática ICIEY2K Fac. de Ingeniería, U.B.A. – Bs. As.

[23] Salcedo Lagos P. (2002) Revista Ingeniería Informática, edición 6, http://www.inf.udec.cl/revista/edicion6/psalcedo.html

[24] Sancho, L. (2002) *Sistemas Tutores Inteligentes: Una alternativa para el uso de computadoras en educación*. Education Net. Red Global de Educación a Distancia. (DistEdNet) Universidad Estatal a Distancia. Consultado en www.uned.ac.cr/biblioteca/global/ensenanza/instruccion/articulos/sistemas.html

[25] Skinner B. F., (1958): *Teaching Machines*, Science, publicado en 1958

[26] Vadillo Zorita J. A., Díaz de Ilarraza A., Fernández I., Gutiérrez J. e Elorriaga J. A. *Explicaciones en Sistemas Tutores de Entrenamiento: Representación del Dominio y Estrategias de Explicación* Departamento de Lenguajes y Sistemas Informaticos Universidad del País Vasco España Consultado en: http://www.niee.ufrgs.br/ribie98/CONG_1994/volume_II/C30/II_289_309.html

El Proceso de Accesibilidad de un Portal Educativo: su Desarrollo y sus Consecuencias

Lic. Javier F. Diaz,

Lic. Ivana Harari y

Lic. Paola Amadeo
[javierd, iharari, pamadeo] @info.unlp.edu.ar
Laboratorio de Investigación de Nuevas Tecnologías LINTI. Facultad de Informática.
Universidad Nacional de La Plata UNLP. Buenos Aires. ARGENTINA.

Abstract

The educational use of Internet is very important for all society levels. It is a global tool for real time queries; it is a framework for knowledge discussion, information interchange and socialization, without geographic and time restrictions.
But users with disabilities, users who suffer vision, comprehension or physical problems, that should take advantage of all the powerful of Internet; they finally, encounter a new digital barrier which makes them impossible to access the Web information.
The educational Web sites more over, should impart politics that improve the access of the Web; they should follow rules for universal Web design, making the site available to a larger audience without arbitrary limits.
The objective of this paper is to convey the experience of the accessibility process developed to the Web site of the Informatics Faculty at La Plata University, where the international standards of accessibility and quality of use were followed and applied. Also, the difficulties of carrying out this process will be given, like the problematic of modifying a great part of the site code in order to reach a product for all.

Keywords: W3C, Web design, Universal design, Accessibility guidelines

Resumen

El uso educativo de Internet, es de gran relevancia en todos los niveles de una sociedad. Es una herramienta global de consulta en tiempo real, un marco para la difusión, intercambio y socialización de conocimientos, que trasciende cuestiones geográficas y restricciones horarias.
Pero, para aquellas personas que sufren alguna discapacidad, problemas de visión, de comprensión, de motricidad, que podrían aprovechar al máximo las potencialidades de Internet, en cambio, se enfrentan a una nueva barrera digital que les imposibilita el acceso a esta importante herramienta.
Los sitios educativos más aún, deberían impartir políticas que mejoren el acceso Web, como así mismo, cumplimentar normas de diseño universal, habilitando el uso de la información a la mayor audiencia posible sin límites arbitrarios.
Este artículo tiene por objetivo contar la experiencia de hacer accesible el sitio de la Facultad de Informática de la UNLP, respetando estándares internacionales de accesibilidad y calidad de uso. También, se va a analizar la complejidad de llevar a cabo este emprendimiento, la problemática de alterar gran parte del código del sitio en pos de lograr un producto al alcance de todos.

Palabras Claves: W3C, Diseño Web, Diseño Universal, Guías de Accesibilidad

1 Introducción

La Facultad de Informática de la Universidad Nacional de La Plata, cuenta con un total de 3000 estudiantes, con un ingreso anual de unos 800 alumnos promedio de las escuelas secundarias, su plantel docente está conformado por unos 200 profesionales, personal administrativo y de gestión.
Ante una comunidad tan amplia, el sitio de informática www.info.unlp.edu.ar es uno de los medios de comunicación más utilizados, tanto para los que ya están integrados a este nivel universitario como para aquellos que están fuera del mismo, y se acercan virtualmente a realizar consultas sobre datos de la Facultad.
El sitio de Informática fue totalmente renovado a fines del año 2006, y en él se pueden realizar transacciones académicas, inscripciones, tener la información integrada de materias con sus accesos a la biblioteca, al plan en que se encuentra, al plantel docente, al sitio de la cátedra, provee el acceso a cursos con modalidad a distancia, presenta en forma centralizada las novedades de todos los sectores de la Facultad, entre otras características.
Dentro del proceso de cambio, se incluyó la necesidad de que el sitio fuera accesible y respete las normas de calidad de uso.
Se consideró de suma importancia que un sitio de una entidad universitaria pueda ser utilizado por todos los usuarios, independientemente de las distintas capacidades auditivas, visuales, motrices o cognitivas que posean, independientemente de los dispositivos, tipo de navegador, recursos tecnológicos que utilicen para acceder al sitio [Ref.1].
Este artículo tiene por objetivo contar todo el proceso de accesibilidad realizado al sitio de la Facultad, las normas que se tuvieron en cuenta, las modificaciones realizadas en el sitio que incidieron principalmente en tres aspectos, modificación del código, codificación de páginas alternativas con contenido redundante pero en formato textual, y alteración de la visualización del sitio original.
También se analizaron los costos y beneficios de llevar a cabo este emprendimiento, es decir la complejidad de alterar gran parte del código del sitio en pos de lograr un producto al alcance de todos.

2 Descripción del sitio de Informática

El sitio de Informática [Ref.2] constituye la puerta de entrada a sectores muy importantes de la Facultad como secretarías y laboratorios, que ya cuentan con sus sitios particulares. A través del sitio de Informática se acceden a más de diez sitios Web pertenecientes a diferentes sectores de la Facultad. Ejemplo de ellos son la Secretaría de Extensión [Ref.3], Postgrado [Ref.4], Biblioteca [Ref.5], Laboratorios de Investigación LINTI, LIDI y LIFIA [Ref.6].
El sitio fue totalmente renovado a partir de septiembre de 2006, y se trabajó principalmente en cuatro puntos: Centralización de la información, Nuevos servicios, Optimizar el mantenimiento y respetar las normativas de accesibilidad de la W3C [Ref.7].
Con respecto a la Centralización de la información: se trabajó sobre la integración de la información y facilitar el acceso a la misma. Esto se debe a que la información estaba muy dispersa entre los sitios particulares a cada sector de la Facultad. Para contrarrestar esto, se realizaron páginas que resumen las actividades de los diferentes sectores de la Facultad, páginas introductorias sobre los sitios Web propios de las Secretarías y Laboratorios que le explican al usuario qué va a encontrar y qué información se va a detallar en los mismos, con un sector de novedades de toda la Facultad, con información centralizada de las materias donde se integran distintas fuentes de información, como ser el acceso al plan al que pertenece y sus correlatividades, el acceso al programa de la misma, a la biblioteca virtual de dicha materia, al sitio de la cátedra y de sus profesores. También, el sitio

provee paneles de Acceso Directo a la información más consultada y en pos de no perder al usuario entre los diferentes sitios Web que existen en nuestra Facultad, se brindó un marco integrador siempre presente, en donde se visualizan internamente los contenidos de dichos sitios particulares. Para proveer consistencia visual en todas las partes del sitio, se trabajó sobre hojas de estilo.
Con respecto a sus nuevos servicios: fundamentalmente, se trabajó en un sistema de búsqueda para todo el sitio de la Facultad, incluyendo todos los sitios de las distintas entidades y sectores que nuclea nuestra entidad. Se realizó, la Comunidad Informática con un directorio de todos los integrantes de la Facultad, se agregaron Barras de Navegación propias al sitio, para imprimir, guardar y moverse sin necesidad de recurrir a las funciones del navegador.
Con respecto a optimizar el mantenimiento, se trabajó sobre un sitio paralelo para la configuración dinámica, con acceso restringido a las autoridades de la Facultad. A través de este sitio se puede mantener dinámicamente y en línea el panel de Novedades, de la Comunidad Informática, de las Materias con toda su información relacionada, de las URLs utilizadas para acceder a los distintos sitios Web, permitiendo creación, modificación y eliminación de datos, cambios que se reflejan al instante en el sitio de la Facultad.
También como unos de los objetivos fundamentales de este proceso de cambio, fue profundizar sobre las normativas impartidas por la W3C, exigiendo que el sitio tuviera en cuenta cuestiones de accesibilidad, calidad de uso respetando los estándares y recomendaciones brindadas por dicho consorcio internacional [Ref.7].
En las siguientes secciones, se van a explicar los detalles del proceso de Accesibilidad que se efectuó sobre el sitio de Informática, presentando el trabajo realizado, los problemas que surgieron y los mecanismos de solución empleados.

3 ¿Por qué hacer accesible el sitio de Informática?

Son muchas las motivaciones que llevaron a hacer accesible el sitio de la Facultad. Una de las principales razones, es por una cuestión de inclusión social. Es decir, permitir que cualquier persona pueda utilizar el sitio, independientemente de problemas visuales, auditivos o motrices que pueda sufrir, sin restricciones respecto al tipo de navegador, tipo de conexión y dispositivos que pueda utilizar.
No se puede diseñar un sitio Web pensando únicamente en las condiciones óptimas de los usuarios, un sitio Web que funcione únicamente a través del mejor enlace a Internet, en la última versión del navegador, con el mejor monitor, que se acceda sólo a través del ratón y por usuarios con todos sus aspectos fisiológicos intactos. Es una situación muy común en muchos sitios Web, pero que es incomprensible si se trata de un sitio educativo.
Otra de las razones tiene que ver con el punto de vista educativo. Desde 1997 se introdujo en nuestra Facultad y dentro de la materia de Diseño Centrado en el Usuario, el concepto de Accesibilidad [Ref.8]. A la par de transmitir esta temática a los alumnos, es imprescindible aplicar estos conceptos, principalmente en el sitio Web que provee la Facultad, así demostrar que se tiene una línea de conducta coherente con los conocimientos que se infunden dentro de la Institución.
Además, se tuvo en cuenta la cuestión legal. Aunque en Argentina aún no hay normas legales que defiendan la accesibilidad, en varios países ya impartieron leyes, normas o reglamentos que colocan a la Accesibilidad en la Web como una obligación legal. Ejemplos de esto, se encuentra la Sección 508 de Estados Unidos, realizada en 1986 pero modificada en los años 92' y 98' [Ref.9] y la Ley 34/2002 [Ref.10], de servicios de la sociedad de la información y de comercio electrónico (LSSICE), que establece que las Administraciones Públicas tomen las medidas necesarias para que la información disponible en sus respectivas páginas de Internet, sean accesibles. En Japón, se

encuentra el programa e-Japan Priority Policy Program, confeccionado en el 2001 [Ref.11], en Italia se encuentra la legislación "Stanca Act", la BITV de Alemania, entre otros.

4 El Proceso hacia la Accesibilidad

Los motivos esgrimidos en la sección anterior, fueron suficientes como para incluir dentro del transcurso de renovación del sitio de Informática, el proceso de Accesibilidad.
El equipo de trabajo estaba constituido por dos desarrolladores Web, una profesional experta en contenidos y una profesional con experiencia en Diseño de Interfaces del usuario.
Los tiempos disponibles eran muy escasos, se tenía previsto sólo un mes de trabajo en el sitio, para la etapa puntual de accesibilidad. Se trabajó durante el mes de enero de 2007, debido a que en esa fecha hay receso lectivo y el sitio es poco utilizado. A partir de febrero, donde el uso del sitio es mayor, el proceso de Accesibilidad debía estar culminado.
Se coordinó y organizó un plan de trabajo estructural, que consistió de las siguientes etapas: primero lograr el entendimiento y conciencia del grupo de trabajo sobre el concepto de Accesibilidad, luego siguieron las etapas de definición de normas a tener en cuenta, selección de herramientas a utilizar que validen la accesibilidad, análisis de los errores y mecanismos de solución.
Como etapa final, se incluyó análisis de los resultados y de las posibles consecuencias o alcances que da lugar un sitio accesible.

4.1 Construyendo el concepto de Accesibilidad

Al principio se trabajó con subjetividad dentro del equipo de trabajo, es decir, se debatió lo que los desarrolladores Web entendían por accesibilidad, cada uno expresó sus opiniones, cuáles eran los objetivos, qué se pretendía de un sitio accesible, cuál era el resultado que esperaban. Se sometió al equipo de desarrollo a experimentar diferentes perfiles de usuarios, a "actuar" como si fueran personas con distintas discapacidades, con escasos recursos tecnológicos que simplemente quieren llevar a cabo tareas elementales en el sitio.
Los mismos programadores y de acuerdo a este trabajo experimental, confeccionaron y estipularon objetivos particulares, que pretendían del sitio. Por ejemplo, que se pueda navegar sólo con teclado, que haya acortamientos con ALT, que funcione correctamente con la tecla TAB, que el contenido del sitio se visualice bien aunque se deshabiliten las imágenes, que ande en distintas resoluciones de monitor y en conexiones lentas, que funcione con un lector de pantalla, entre otras cuestiones.
Una vez que se trabajó con esta etapa de concientización y de construcción en conjunto del sentido de la Accesibilidad, de sus alcances e importancia, se comenzó a dilucidar en los mecanismos de "cómo lograrla". Se los introdujo al mundo de las normas y directivas de accesibilidad de la W3C, denominadas WCAG 1.0 [Ref.7] y a los programas existentes para verificar y validar el cumplimiento de las mismas.
De esta manera, el equipo de trabajo profundizó sobre las recomendaciones y el soporte que se brinda desde la W3C-WAI, sobre la factibilidad de llevar a cabo este proceso de Accesibilidad, y concluyó con la definición de los principales objetivos que había que lograr en el sitio.
Se estipuló que el sitio debía cumplir con el proceso de validación de Accesibilidad, de Calidad de Uso de XHTML 1.0 y de Hojas de estilo CSS 2.0, y en ese proceso se encaminaron.

4.2 Las normas o recomendaciones que se tuvieron en cuenta

Las pautas de accesibilidad se definen según el World Wide Web Consortium (W3C) de la siguiente manera: "Estas pautas explican cómo hacer accesibles los contenidos de la Web a personas con discapacidad. Al respetarlas, se hará la Web más asequible también para todos los usuarios, cualquiera que sea la aplicación de usuario que esté utilizando, como ser, navegador de sobremesa, navegador de voz, teléfono móvil, PC de automóvil, o las limitaciones bajo las que opere. Seguir estas pautas ayudará también a que cualquier persona encuentre información en la Web más rápidamente."

Las Guías de Accesibilidad para el Contenido Web -WCAG 1.0-, presentan 14 pautas que constituyen los principios generales del diseño accesible. Cada pauta detalla uno o más puntos de verificación que constituyen principios más específicos de Accesibilidad. Cada punto de verificación de las pautas, tiene asignada una determinada prioridad, según el nivel de importancia que tiene el cumplimiento del mismo, dentro del proceso de accesibilidad**.**
Existen 3 tipos de prioridades. Los puntos de verificación que "deben" cumplirse son los de prioridad 1, los que "deberían" cumplirse son de prioridad 2 y los que "podrían" cumplirse son los de prioridad 3.
La W3C otorga tres niveles de clasificación: clasificación nivel "A", en donde todos los puntos de verificación de prioridad 1 fueron satisfechos, de nivel "AA", en donde todos los puntos de verificación de prioridad 1 y 2 fueron satisfechos y los de nivel "AAA" en donde el sitio cumple con todas las pautas y todos sus puntos de verificación.

4.3 Herramientas de evaluación de Accesibilidad utilizados

Las herramientas de evaluación de accesibilidad pueden ser utilizadas en todas las etapas del desarrollo del sitio Web. Puede utilizarse en la etapa inicial del diseño, en donde las normas de accesibilidad podrían ser consideradas como nuevos requerimientos a considerar en el desarrollo del sitio. En el caso específico del sitio de Informática, cuando el mismo fue sometido al proceso de Accesibilidad, ya se encontraba totalmente desarrollado y estaba en uso, con aproximadamente más de 4000 visitas mensuales. Por lo tanto, en este caso, se utilizaron las herramientas de validación de Accesibilidad en la etapa de evaluación del sitio y sus resultados constituyeron la fuente de información para el proceso de corrección y modificación del mismo.
Las herramientas de evaluación de Accesibilidad presentan diferentes características [Ref.12], que fueron analizadas para poder elegir con cuál de ellas trabajar.
En primera medida, hay que considerar con qué normativas trabaja la herramienta. Si considera las normas WCAG 1.0, las de Sección 508 o sigue los estándares de Japón, de Italia, u otros. Hay que analizar los tipos de tecnologías Web que soporta, si simplemente HTML y XHTML, o aquellas más sofisticadas como SMIL, SVG, o MathML. Es fundamental verificar qué plataforma soporta y cómo se puede integrar dentro del ambiente de desarrollo utilizado, si mediante plug-ins en el navegador [Ref.17] o como utilitarios dentro del editor o sistema de administración de contenido Web.
También, hay que analizar el tipo de soporte que brinda la herramienta respecto al chequeo de errores, si genera reportes, si muestra a través de feedback visual íconos o remarca las etiquetas erróneas sobre la misma página, si permite evaluación paso a paso. En el caso de herramientas más avanzadas, éstas permiten realizar transformaciones sobre el diseño de tu sitio, por ejemplo convertirlo totalmente a modo textual.
Luego de un proceso de análisis, y teniendo en cuenta los requisitos por parte de los desarrolladores Web, los cuales manifestaron sus preferencias por encontrar una herramienta que sea simple de

utilizar, que venga en lenguaje español y de libre licencia, se eligieron las siguientes herramientas para trabajar: se utilizó para la validación de Accesibilidad, la herramienta TAW On Click [Ref.13], el cuál se instala y directamente se encuentra presente en el navegador del Firefox [Ref.14] mediante un icono, mostrando los errores visualmente sobre la misma página que se está analizando. Para la validación de la gramática del sitio y calidad de uso XHTML 1.0, se utilizó el W3C Markup Validation [Ref. 15], y para la validación de las plantillas de estilo CSS 2.0, se trabajó con la herramienta JigSaw [Ref.16].
Inicialmente, los desarrolladores Web habían expresado en forma clara su preocupación y disconformidad por tener que trabajar con herramientas nuevas en tan poco tiempo, temiendo tener problemas de instalación, utilización o de no poder comprender la información que generan las mismas, aumentando así la complejidad de todo el proceso de Accesibilidad. Esta sensación se revirtió rápidamente al trabajar con las herramientas seleccionadas, por la simpleza y la valiosa asistencia que las mismas ofrecen.

4.4 Análisis de los errores de accesibilidad del sitio de Informática original

El sitio de Informática, en su primera prueba de accesibilidad realizada frente al validador TAW (Figura 1), arrojó la siguiente tasa de problemas, identificados por la herramienta como "errores automáticos", pues son etiquetas HTML que provocan un fallo en la accesibilidad:

- 15% de errores de Prioridad 1 (11 en total)
- 50% de errores de Prioridad 2 (39 en total)
- 35% de errores de Prioridad 3 (27 en total)

Respecto a los problemas manuales, "Human Review", que deben ser verificados por la persona evaluadora, los más destacados fueron que los textos alternativos proporcionados sean los adecuados y el sitio se visualice en forma correcta sin hojas de estilos. Estas revisiones manuales no afectan a la clasificación.

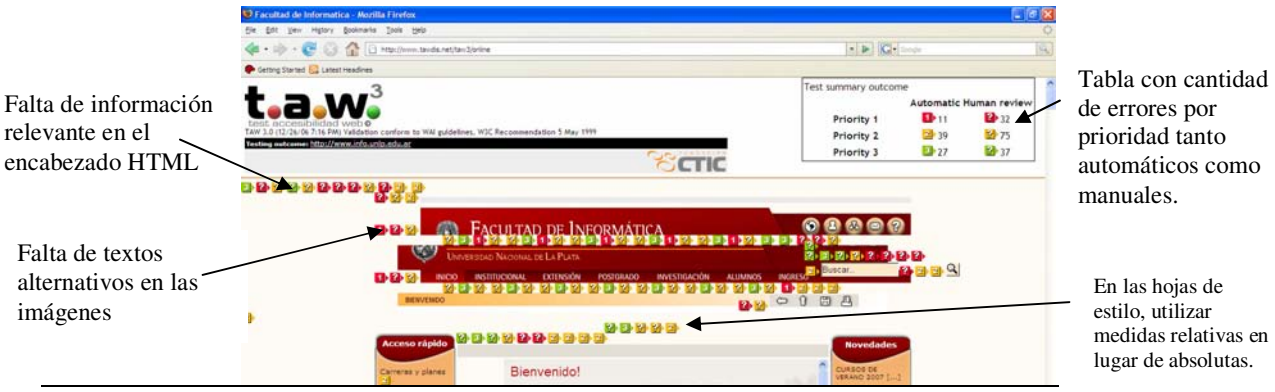


Figura 1: Pantalla del validador TAW con la visualización de los errores encontrados

Entre los errores más cometidos en todas las páginas del sitio podemos mencionar:

- No utilizar el &accute para los acentos y códigos para las ñ
- Falta de textos alternativos para imágenes, íconos e hipervínculos.
- Falta de información de resumen en las tablas.
- Falta de uso de directivas básicas de HTML, como <H1> <H2> <H3>, que prácticamente en la versión original no se utilizaban.
- Uso de valores absolutos para ubicar imágenes, particularmente las barras de menús de los extremos.
- Falta de títulos en los frames utilizados
- Falta de información importante en el HEADER de HTML, por ejemplo el idioma utilizado, la versión HTML y la codificación ISO utilizada en el documento.

- Uso de formatos no estándares estipulados por la W3C, por ejemplo el formato .doc como único formato de un documento.
- La tecla TAB no estaba implementada en forma correcta.

El sector con mayor dificultad para adaptar fue la página inicial. Entre otros problemas podemos mencionar:

- El uso de íconos e imágenes que enriquecen la visualización no garantizan que se desplieguen de la misma manera en todos los navegadores y con diferentes tamaños de pantalla.
- La correcta visualización de las páginas configurando el navegador para navegar sin imágenes. En forma similar sin hojas de estilos.
- La configuración del acceso por teclado para las opciones de menú. Los navegadores implementan el accesskey de diferente manera, por ejemplo en Firefox para acceder a un link es necesario presionar en forma conjunta las teclas Alt + Shift + tecla resaltada.
- Respetar la combinación de colores estipulada por la Facultad. Esta combinación no siempre respetan las normas.
- Las páginas dinámicas que utilizan muchas tablas, como Novedades o Planes de Estudio.

4.5 Mecanismos de solución para implantar la accesibilidad en el sitio

En el proceso de implementación de la accesibilidad se trabajó también con la herramienta TAW para escritorio. Este sistema es más completo que su versión Web, facilitando la identificación y corrección de errores con distintos grados de prioridad (Figura 2).

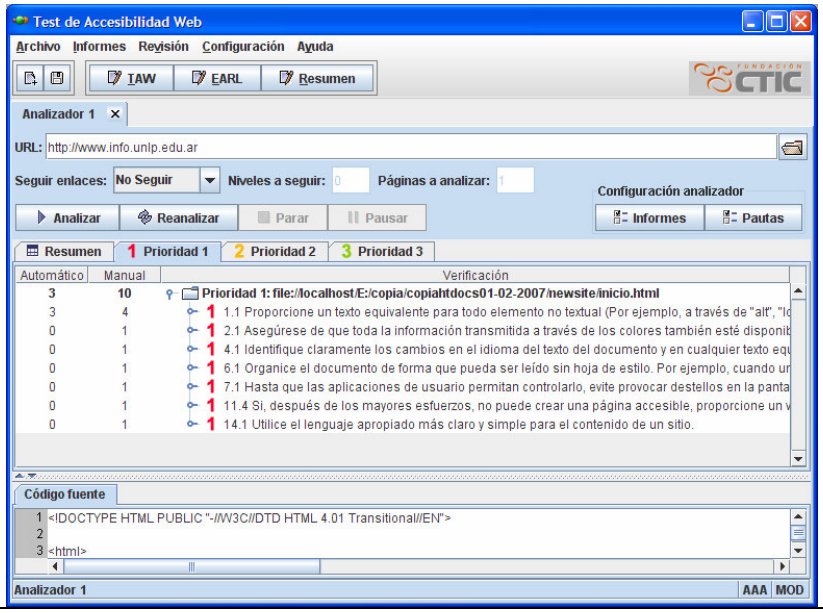


Figura 2: Pantalla de TAW para escritorio

Respecto a las pautas de Prioridad 1, que es necesario respetar para lograr la clasificación A, la mayoría de los problemas se solucionaron mediante la corrección y modificación del código HTML del documento. Se comenzó agregando texto alternativo a todos los íconos e imágenes utilizados. En la Figura 3, se visualiza cómo se muestra el sitio deshabilitando las imágenes del navegador tanto en su versión no-accesible como en la accesible. Se puede notar que las imágenes se muestran con una cruz [X] en la versión no accesible, mientras que en la versión accesible en lugar de las imágenes aparecen explicaciones sobre las mismas.

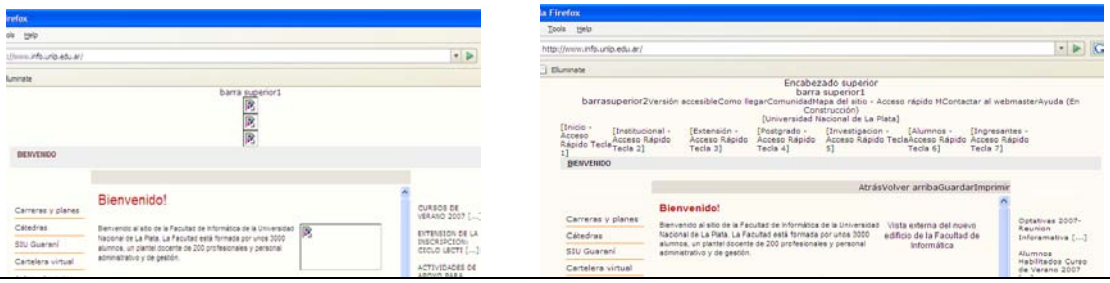


Figura 3: El sitio original y el sitio accesible deshabilitando las imágenes desde el navegador

También se corrigió el encabezado del documento HTML para incluir información relevante, tal como idioma utilizado, versión de HTML, codificación ISO y demás datos propios de esta sección.
Los accesos directos y la configuración de la tecla TAB fue una tarea que demandó tiempo y decisiones de diseño acerca de la forma más adecuada de proveer esta información y que le sea útil al usuario. El principal inconveniente fue la cantidad de hipervínculos y opciones de menú que parte de la portada. Es así como los accesos rápidos de la barra de menú principal se decidió implementarlos a través de números y utilizar el texto alternativo para notificar a los usuarios de esta funcionalidad, como se muestra en la Figura 4.

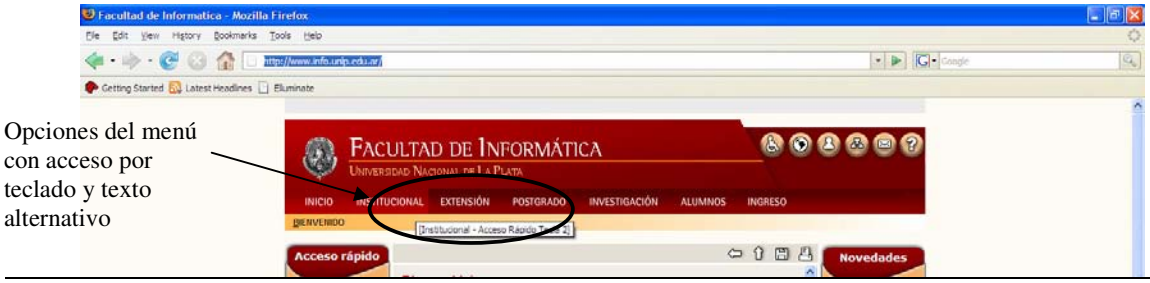


Figura 4: Texto alternativo y numeración en los menúes

En las tablas se incluyó una descripción clara y concreta de las mismas, tanto en sus filas como columnas. Sin embargo, lograr una versión accesible de las páginas dinámicas, como Novedades y Cátedras, no fue una tarea sencilla, ya que toda la información se mostraba en tablas que se construyen a partir de información almacenada en una base de datos. Es así como se decidió implementar las páginas dinámicas reemplazando dichas tablas con listas, encabezados y capas en nueva páginas alternativas y accesibles.
En la figura 5 se muestra la página de Cátedras en sus dos versiones, la original que es puramente visual y la accesible, que es mayormente textual.

Figura 5: Página dinámica en sus versiones original y accesible.

Respecto a las pautas de Prioridad 2 y 3, necesarias para lograr la clasificación AA o AAA, se modificó el código para lograrlas, pero las que implicaba un cambio en la visualización, fueron resueltas parcialmente. Se retocaron colores de fondo y letras, y hubo reubicaciones de

componentes. Pero los cambios más importantes que era necesario efectuar en la pantalla inicial, donde se usan valores absolutos e imágenes en los paneles funcionales, no se pudieron concretar. Fue bastante conflictivo por las reticencias por parte de las autoridades, quienes solicitaron no retocar la visualización del sitio, principalmente en la portada. De esta manera la misma quedaba solamente con un nivel A de accesibilidad.
Se decidió entonces acudir al desarrollo de páginas alternativas, se proporcionó un hipervínculo a una portada alternativa que use tecnologías W3C, que sea completamente accesible, o sea que alcance el nivel de conformidad AAA y tenga contenido equivalente a la portada original. A esta versión accesible se accede a través de la página principal con el ícono  o directamente a través de la URL: http://www.info.unlp.edu.ar/accesible/ (Figura 6).

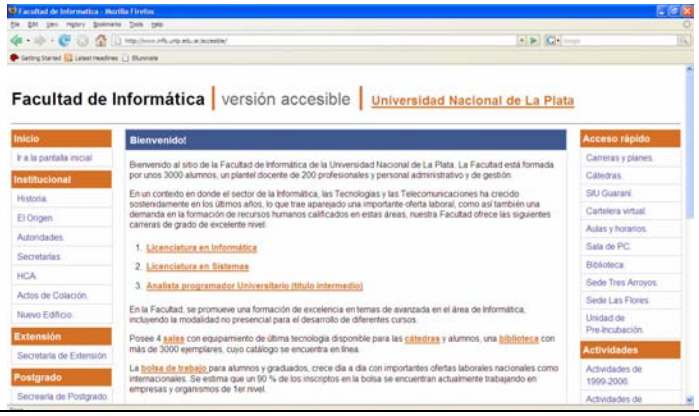


Figura 6: Portada accesible nivel AAA.

Finalmente, se puede resumir los mecanismos de solución empleados con el fin de reducir la tasa de errores y lograr la accesibilidad. Estos son: modificación del código, realización de páginas alternativas con contenido redundante pero en formato textual, y alteración de la visualización del sitio original.
Con respecto a la modificación del código, el 75% de los errores fueron solucionados de esta manera. La realización de páginas alternativas con contenido redundante, fue un recurso empleado en sectores muy importantes del sitio como ser en las páginas dinámicas de Novedades y Cátedras, y en la portada del mismo. De esta manera, el sitio provee versiones altamente visuales que alcanzan accesibilidad A y versiones alternativas donde predomina el texto con accesibilidad AAA.
La modificación de la visualización, fue uno de los mecanismos menos utilizados y se trabajó en este punto con muchas restricciones.
En la siguiente sección, se van a analizar los resultados de este proceso de accesibilidad y se van a contemplar los alcances o consecuencias de haber logrado este importante objetivo.

5 Consecuencias del Proceso de Accesibilidad

El proceso de accesibilidad produjo grandes satisfacciones, como consecuencia de la tarea de por si laboriosa de adaptación, se obtuvo un producto que respeta estándares internacionales a fin de lograr la inclusión digital. Se logró alcanzar la clasificación AAA de Accesibilidad, el de calidad de uso de XHTMl 1.0 y el de las hojas de estilo CSS 2.0.
La tasa de errores bajó notablemente, de inicialmente 11 de errores de tipo 1 (15%) se disminuyó a 0, de 39 errores de tipo 2 (50%) se disminuyó a 7 y de 27 errores de tipo 3 (35%) se logró disminuirlo a 5.

El sitio funciona correctamente sin el uso del ratón, mediante las teclas TAB, ALT y <Enter>, deshabilitando las imágenes y las plantillas de estilos.
También, se hicieron pruebas con el software de lectura de pantalla JAWS [Ref.18], muy utilizado por la comunidad invidente argentina. El objetivo fue analizar la usabilidad y navegabilidad del sitio accesible utilizando este producto. Fue increíble como se notó el buen funcionamiento ni bien se ingresó al sitio, pudiendo accederse a todas las opciones y a la información. En base a esto, se tuvieron que realizar modificaciones menores como adelantar algunas opciones para que el lector las encuentre más rápido y modificar algunas combinaciones de teclas, que se usan en ese entorno, como por ejemplo CTRL-D que en el sitio de Informática se accede a Cátedras y en JAWS se usa para acceder a las direcciones urls del sitio.
Finalmente, se comprobó también la estrecha relación entre sitios accesibles con sitios para Web móvil, se pudo observar el sitio accesible de la Facultad desde un teléfono celular en forma inmediata.
Estas pruebas con resultados altamente satisfactorios, gratificaron enormemente al equipo de trabajo, comprobando que los esfuerzos y la preocupación de tener que modificar y retocar todo el código de un sitio para lograr accesibilidad, valen la pena para todos.

6 Conclusiones

Dentro de un proyecto de accesibilidad en la Web, promovido desde la Facultad de Informática de la UNLP, se propuso hacer accesible el mismo sitio de la Facultad, aplicando las normativas de diseño universal y calidad de uso.
Se explicó todo el trabajo realizado en este proceso de Accesibilidad, se explicaron las modificaciones realizadas en el sitio, en donde se retocó toda su codificación, se generaron nuevas páginas alternativas con información redundante y hasta tuvo cambios a nivel visual.
El nivel de frustración que experimentaron los desarrolladores Web al analizar por primera vez el sitio desde las herramientas de evaluación de Accesibilidad, en donde arrojaron una tasa considerable de errores, fueron desapareciendo en cada día de trabajo. La selección y uso de herramientas adecuadas para cada etapa del proceso simplificó enormemente la tarea.
Finalmente, el sitio actual cuenta con la calificación AAA de la W3C, el de calidad de uso de XHTMl 1.0 y el de las hojas de estilo CSS 2.0
Además, el proceso de accesibilidad no culmina en su totalidad, hasta poder descubrir las ventajas y los alcances que conlleva el haberlo logrado. Se realizaron pruebas manuales, donde se comprobaron el correcto funcionamiento del sitio en condiciones no óptimas, como ser sin mouse, inhabilitando las imágenes, las plantillas de estilo, en versiones de navegadores más viejas y en distintas resoluciones de monitor. Se probó el sitio accesible desde productos de adaptación para invidentes como el JAWS, obteniendo resultados altamente satisfactorios y hasta se tuvo una interacción con el sitio en forma inmediata a través de un celular.
También, como consecuencia de este proceso se dio lugar a la apertura de nuevas líneas de acción como ser la evaluación del sitio frente a usuarios ciegos, organizando en un futuro muy próximo testeos de usabilidad con ellos. También, se plantea profundizar sobre la incidencia de la accesibilidad dentro de la Internet Móvil y comenzar con la adecuación de la interfaz accesible del sitio de Informática para el acceso desde diferentes dispositivos.
Se intentó aportar en este artículo algo más que la transformación misma de un sitio no accesible a un sitio que ya lo es, es el hecho de introducir en cada paso realizado, los temores, dilucidaciones y conclusiones que se manifestaron dentro del equipo de desarrollo, en pos de lograr que este proceso de Accesibilidad concluya en forma satisfactoria con sus efectos altamente positivos.

7 Referencias

[Ref.1] Haddad, W & Draxler, A "Technologies for education. Potencials, parameters and propects", UNESCO & Knowledge Enterprise. (2002).

[Ref.2]http://www.info.unlp.edu.ar

[Ref.3] http://www.info.unlp.edu.ar/index.php?mostrar=extension

[Ref.4] http://www.info.unlp.edu.ar/index.php?mostrar=postgrado

[Ref.5] http://www.info.unlp.edu.ar/index.php?mostrar=biblioteca

[Ref.6]http://www.info.unlp.edu.ar/index.php?mostrar=linti,http://www.info.unlp.edu.ar/index.php?mostrar=lidi y http://www.info.unlp.edu.ar/index.php?mostrar=lifia

[Ref.7] http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505

[Ref.8] http://www.info.unlp.edu.ar/actions/download.php?idArchivo=160

[Ref.9] http://www.access-board.gov/sec508/guide/act.htm

[Ref.10] http://www.congreso.es/public_oficiales/L7/CONG/BOCG/A/A_068-13.PDF

[Ref.11] http://www.kantei.go.jp/foreign/it/network/priority-all/index.html

[Ref.12] http://www.w3.org/WAI/ER/existingtools.html

[Ref.13] http://www.tawdis.net/taw3/cms/es

[Ref.14] http://www.firefoxflicks.magnify.net/

[Ref.15] http://validator.w3.org/

[Ref.16] http://jigsaw.w3.org/

[Ref.17] http://www.mozilla.com

[Ref.18] http:// http://www.freedomscientific.com/fs_products/software_jaws.asp

El tema de la calidad en la educación a distancia. Propuesta de un modelo de evaluación en capas *

Dra. Cecilia Sanz[1], Esp. Gladys Gorga[2],Esp. Cristina Madoz[3]
{csanz, ggorga, cmadoz} @lidi.info.unlp.edu.ar

Instituto de Investigación en Informática. LIDI (III LIDI)
Facultad de Informática. UNLP

Abstract

In III LIDI, there exists a subproject related to the topic of computer technology applied to education. Within the context of such project, topics related to quality and, in particular, to quality in distance education are being studied.
This paper presents a review of the concept of quality, in particular quality in higher education. Then, the discussions related to new educational modalities and quality assessment are analyzed. Different opinions of renowned authors in the scope of distance education are presented together with the assessment models they propose.
Finally, a quality assessment model in distance education is proposed, which presents a series of layers or levels to be taken into account. This model is still being developed and will be applied to different cases in our institution, in order to study the difficulties presented at the moment of its implementation.
Keywords: Quality, Distance Education, Quality Model.

Resumen

En el III LIDI existe un subproyecto vinculado al tema de tecnología informática aplicada en educación, en el marco de dicho proyecto, se estudian temas relacionados con la calidad y en particular con la calidad en la educación a distancia.

En este artículo se presenta una revisión del concepto de calidad, en particular calidad en educación superior. Luego, se analizan los debates generados en vinculación con las nuevas modalidades educativas y la evaluación de la calidad. Se presentan diferentes opiniones de autores reconocidos en el ámbito de la educación a distancia, y los modelos de evaluación que proponen.

Finalmente, se propone un modelo de evaluación de calidad en la educación a distancia, que presenta una serie de capas o niveles a tener en cuenta para la evaluación. Este modelo se continúa desarrollando y se aplicará a diferentes casos de nuestra institución, para estudiar las dificultades que presenta al momento de la implementación.

Palabras claves: calidad, educación a distancia, modelos de calidad.

[1] Prof. Adjunto Dedicación Exclusiva. Facultad de Informática. UNLP.
[2] Prof. Adjunto Dedicación Exclusiva. Facultad de Informática. UNLP.
[3] Prof. Adjunto Dedicación Exclusiva. Facultad de Informática. UNLP.

* Esta investigación forma parte del "Proyecto Sistemas de Software Distribuidos. Aplicaciones en procesos industriales, E-government y E-learning, del Instituto de Investigación en Informática LIDI. Subproyecto E-Learning".

1.- Calidad en Educación Superior en general

En la sociedad actual, la educación superior debe atender a la demanda tradicional, pero ahora también surgen demandas de los profesionales incorporados al mercado laboral que requieren de actualización y capacitación permanente, y los adultos mayores que deben renovar constantemente sus conocimientos para responder a los cambios tecnológicos que la sociedad le presenta.

La evolución de esta sociedad se apoya fuertemente en el desarrollo de nuevas tecnologías de información y comunicación, impulsadas por los adelantos de la informática y la telemática, que impactan en todos los ámbitos de la vida en sociedad.

Las instituciones de educación superior deben aprovechar al máximo el potencial de estos recursos estableciendo propuestas innovadoras que contribuyan a alcanzar los objetivos trazados.

Desde siempre, la Educación Superior Universitaria se ha caracterizado por una constante preocupación por la excelencia de su modelo educativo y por la transformación continua que le exige la sociedad en la que está inserta. En este contexto surgen modelos de educación superior que deben incorporar nuevos paradigmas de funcionamiento para dar respuesta a las necesidades actuales.

Por otro lado, es importante tener presente que satisfacer las demandas de esta nueva sociedad, provoca un crecimiento notorio de la oferta de educación superior desde el punto de vista de las entidades que las proveen y de las modalidades que las mismas adopten. Incluso en muchas ocasiones esta oferta está asociada a fuertes intereses comerciales. A partir de esta situación, el tema de la calidad educativa ocupa un lugar central en el diseño de políticas educativas y en la construcción del marco normativo de la Educación Superior.

En la Conferencia Mundial sobre la Educación Superior (1998), se realizó un análisis de la situación de la enseñanza superior que puso de relieve tres prioridades dominantes a las que se debe conceder la máxima atención: la ampliación del acceso sobre la base del criterio del mérito, la renovación de los sistemas e instituciones de educación superior y el fortalecimiento de los vínculos con la sociedad, en especial con el mundo del trabajo [1].

Este afán de renovación de la enseñanza superior se articula en torno a cuatro exigencias fundamentales: la pertinencia, la calidad, la gestión y la financiación, y la cooperación internacional. A continuación se mencionan algunos conceptos o ideas vinculados a la calidad de la educación superior, obtenidas de dicha declaración, de manera tal de dar un marco a esta presentación.

“La calidad de la educación superior es un concepto multidimensional que debería comprender todas sus funciones y actividades: enseñanza y programas académicos, investigación y becas, dotación de personal, alumnos, infraestructura y entorno académico. Ha de prestarse especial atención al progreso de los conocimientos mediante la investigación. Las instituciones de educación superior de todas las regiones han de someterse a evaluaciones internas y externas realizadas con transparencia, llevadas a cabo abiertamente por expertos independientes. Sin embargo, ha de prestarse la debida atención a las particularidades de los contextos institucionales, nacionales y regionales, a fin de tener en cuenta la diversidad y evitar la uniformidad. Se percibe la necesidad de una nueva visión y un nuevo modelo de educación superior, que debería estar centrado en el estudiante. Para alcanzar dicho objetivo, hay que reformular los planes de estudio, no contentarse con el mero dominio cognoscitivo de las disciplinas e incluir la adquisición de conocimientos prácticos, competencias y aptitudes para la comunicación, el análisis creativo y crítico, la reflexión independiente y el trabajo en equipo en contextos multiculturales” [2]

El párrafo anterior muestra un cambio en los modelos educativos tradicionales, buscando centrar la atención en los estudiantes y en los procesos relacionados con la adquisición de conocimientos y desarrollo de competencias vinculadas con la cognición y la metacognición.

En este sentido, resultará necesario que las instituciones de nivel superior fijen claramente los criterios y estándares de calidad a seguir para alcanzar sus fines teniendo en cuenta, principalmente, el papel crítico que desempeñan en la sociedad del conocimiento actual.

Para determinar el conjunto de criterios de calidad de las instituciones de nivel superior que adopten cualquiera de las modalidades presencial o a distancia, será fundamental analizar cuál es el contexto particular en el que se desarrollan, cuáles son los componentes de ese contexto, cuales son los aspectos mas críticos a considerar, cuáles son los actores que formarán parte, qué características tienen esos actores, qué necesidades y demandas, etc. porque todas estas consideraciones en relación con los objetivos que se persiguen, determinará los caminos que seguirá el proyecto educativo en cuestión en un marco de calidad sostenida y mejora continua.

2.- Calidad en EAD. Debates no resueltos

Como se analizó en el apartado anterior, el cambio que se viene originando en el ámbito de la Educación superior, trae aparejado un replanteo de los actuales criterios que buscan asegurar la calidad de los modelos de las instituciones educativas [10] [11].

Para indagar sobre la calidad deben establecerse criterios, estándares, indicadores, etc., que puedan ayudar a medir o a valorar, bien sean las ofertas del mercado de formación, bien nuestros propios diseños y productos formativos.

Los criterios mediante los que se emitirán los juicios de valor serán el tema clave de las evaluaciones y lo que marcará, en realidad, el nivel de la calidad de la educación ofrecida.

Al momento de definir los criterios a tener en cuenta, necesariamente debemos considerar el contexto, y los objetivos que pretenden lograrse. Seguramente los instrumentos que utilicemos para recoger datos acerca de la calidad, contemplarán estos aspectos. García Aretio afirma, "…los diferentes grupos de poder o de pensamiento, podrán considerar una calidad alcanzada, en función del planteamiento previo del que se partiese y de los objetivos pretendidos."

Como mencionan Barberá et al [3], referido a la evaluación de la calidad en sistemas educativos, "…el primero de los errores es la copia casi perfecta de los modelos de calidad de empresa aplicados a la educación; ni la forma ni el contenido puede traspasarse de la manera que se está haciendo a entornos educativos, por no hablar de sus resultados. Modelos de calidad total (TQ, EFQM, entre otros) más basados en la gestión que en los procesos de enseñanza y aprendizaje son aplicados en la educación presencial y empiezan a extenderse en la educación a distancia sin previa reflexión, simplemente traduciendo de manera intuitiva, en la mayoría de los casos, los términos y los conceptos empleados en entornos empresariales"

Refiriendo nuevamente a García Aretio, menciona que buena parte de las diferentes propuestas y modelos sientan sus raíces en el Modelo Europeo dado que pretenden ofrecer un modelo de evaluación de la calidad que tenga en cuenta la "totalidad" (vinculado a la visión del mundo empresarial como hemos ya dicho). El EFQM (European Foundation for Quality Management), centra su interés, especialmente, en la satisfacción de sus clientes, basando la misma en la mejora continua, medición y máxima atención a los procesos, el trabajo en equipo, y la responsabilidad de cada uno. El modelo supone una estrategia global que aspira a lograr la calidad de cada aspecto de funcionamiento de una organización [4].

Como se ha mencionado en la sección inicial, es difícil encontrar una única mirada sobre la calidad, y así existen modelos que ponen el énfasis en la satisfacción de los estudiantes (siguiendo la visión

empresarial de satisfacción del cliente), algunos en el impacto social, y otros en la efectividad de los procesos.

Pero, en este momento nos surgen los siguientes interrogantes: si los modelos de calidad en educación superior deberían diferenciarse de los modelos aplicados al ámbito empresarial, ¿cómo deberían pensarse y dónde deberían poner el foco, cuáles serían los criterios a considerar? Por otra parte, ¿es necesario generar modelos de calidad diferentes para distintas modalidad? ¿La educación a distancia requiere de un modelo diferenciado de la educación presencial?

Según Sangrà (2002) existen dos tendencias básicas en cuanto a la determinación de la calidad de la Educación a Distancia: quienes la consideran un instrumento auxiliar de la presencialidad y quienes la conciben como una entidad con especificidad propia. La tendencia predominante que se afirmará cada vez con mayor fuerza es la segunda. Sangrà, sostiene que la EAD debe evaluarse de acuerdo a criterios en cuanto a la oferta formativa, y su pertinencia en relación a necesidades sociales y del mercado laboral; la organización y tecnología; los materiales; la docencia y la creación de conocimiento [5].

Sin embargo, en una entrevista al Dr. Ernesto Villanueva, al ser consultado acerca de su opinión sobre si es necesaria una acreditación específica de las carreras en modalidad a distancia, afirma que no acuerda con la postura de tener un modelo de acreditación específica [6].

Como vemos, hay puntos encontrados también en este sentido. Nos quedaremos con aquellos modelos que son específicos para la educación a distancia.

3.- Algunos criterios a considerar para evaluar un sistema, programa o propuesta educativa a distancia. Reseña de la opinión de diferentes autores

Mencionaremos a continuación una serie de criterios propuestos por algunos autores reconocidos, que nos permitirán analizar la calidad de una propuesta o sistema de educación a distancia, y dan un marco a este trabajo.

Moore, menciona algunos de los criterios a tener en cuenta en la evaluación de la calidad de un sistema educativo a distancia [7]. También abordaremos algunos mencionados por otros autores como Vergara, García Aretio, etc.

Criterios mencionados por Moore

a. Inscripción de los alumnos: ¿la tasa de inscriptos a cursos ofrecidos por la institución o programa se mantiene constante o crece año a año? Esto puede indicar una buena reputación, satisfacción de los estudiantes, que llevan a una difusión positiva de la institución.

b. Logros de los estudiantes: es un aspecto difícil de medir. Dado que las evaluaciones varían entre una y otra institución, y dependerán también de los evaluadores. En algunos casos se trata de evaluar el desempeño de los graduados en su ámbito laboral.

c. Satisfacción de los estudiantes: es fácil de medir. En general, los estudiantes ya están acostumbrados a la práctica de evaluar un curso luego de su conclusión, analizando su contenido, organización, tutores, materiales instruccionales, y sistema tecnológico de soporte. Esto provee al menos un chequeo mínimo de la calidad, en cuanto a la percepción de los estudiantes. Esto no mide la efectividad en términos de aprendizaje, y tampoco valida la relevancia del contenido enseñado.

d. Satisfacción de la institución: es importante de medir pero tiene ciertas limitaciones. La institución puede ver hasta que punto las estrategias de enseñanza y los materiales parecen ser efectivos, y si la atención de los estudiantes ha sido adecuada. Los datos de evaluación de la institución pueden ser poco objetivos.

Estos criterios están vinculados a modelos de calidad que ponen el foco en los resultados y el impacto producido, y no focalizan como prioridad criterios vinculados al proceso en sí mismo.

García Aretio [8] propone un modelo que entiende como integrador y de desarrollo y control de la calidad total, vinculado al contexto, metas, entradas, procesos, resultados, y mejoras. A continuación se mencionan los criterios a considerar en este modelo:

a. Funcionalidad: coherencia entre objetivos y resultados educativos respecto del sistema de valores, expectativas, necesidades culturales y socio-económicas de una Comunidad. Si resumimos, pensamos que esto refiere a una adecuación de los objetivos y resultados de la institución para con las necesidades y realidad contextual.

b. Eficacia o efectividad: coherencia entre metas educacionales y los resultados obtenidos.

c. Eficiencia: coherencia entre entradas, procesos, medios y los logros o resultados educativos. Por ejemplo, adecuación y validez de las estrategias de intervención, tiempos, recursos instructivos y curriculares a las metas educativas.

d. Disponibilidad: coherencia entre las metas y objetivos propuestos institucionalmente y los recursos humanos, materiales y económicos de que pueden disponer para iniciar el proceso.

e. Información: coherencia entre los resultados obtenidos y las propuestas de mejora que se ofrecen en el informe pertinente.

f. Innovación: coherencia entre el catálogo de mejoras precisas (reforzamiento de puntos fuertes, y corrección de los débiles) para el logro de las metas, y la decisión de innovar y revisar metas, entradas y procesos.

Finalmente, se ha analizado el trabajo de Zamora [9] que referencia una serie de buenas prácticas, para asegurar la calidad, vinculadas al compromiso institucional, costos, normas legales, diseño instruccional, etc.

En la siguiente sección se presentará la propuesta diseñada por los autores de este trabajo para evaluar la calidad de un sistema / programa de educación a distancia, que integra algunos de los elementos previamente vistos.

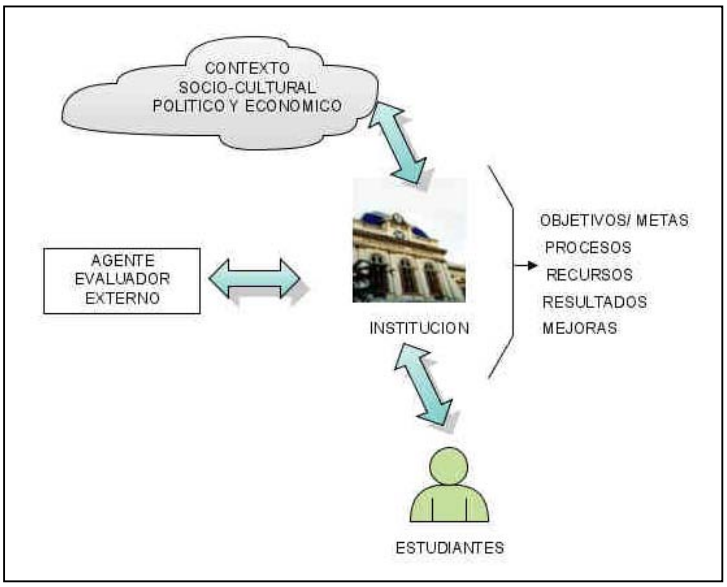


Figura 1. Componentes del sistema a tener en cuenta en el modelo de evaluación de calidad que se propone en este artículo

4.- Propuesta de un posible modelo para analizar la calidad de un sistema de educación a distancia

Teniendo en cuenta, los criterios y dimensiones mencionados por autores reconocidos en el área de educación a distancia, hemos abordado el diseño de un modelo de análisis de calidad en capas, tal como se describirá a continuación. En la primera capa se considerarán criterios generales, en la segunda algunos indicadores, y en la tercera, se buscará dar algunas medidas referidas a los indicadores de la segunda capa. A medida que se descienda en las capas, se ganará en especificidad respecto de un sistema de educación a distancia.

Al igual que muchos otros autores, creemos que la Institución está inmersa en un cierto contexto socio-cultural, político y económico, que presenta determinadas demandas, necesidades, conflictos frente a las universidades y otras instituciones de educación superior (IES). En general, existen indicadores que la IES tiene en cuenta, para estudiar lo que el contexto requiere o hacia donde está evolucionando.

Si bien los estudiantes forman parte de este contexto del que hablamos, le hemos dado un rol central como componente de este sistema, ya que tienen intereses específicos frente a la institución, forman parte de ella, en muchos casos eligiéndola, y esperando determinadas acciones o resultados de la IES. Puede aportarle información acerca de sus necesidades, expectativas. Al mismo tiempo, la IES impacta sobre el alumno de alguna manera (a partir de sus metas/objetivos). El alumno percibe la institución y puede dar opinión acerca de ésta.

Los Agentes de Evaluación externa son otro componente fundamental, que toman información de la institución, y dan a la misma un "feedback" específico para su evolución en la calidad educativa.

La institución, para su funcionamiento, establece cuáles son sus objetivos/metas, define procesos para alcanzar dichos objetivos, dispone recursos de distinta índole (que organiza y distribuye con ciertas estrategias), busca resultados que son evaluados mediante procesos internos y/o externos, que finalmente, permiten establecer puntos de mejoras. Entendemos que si la institución logra establecer la funcionalidad, disponibilidad, eficacia, eficiencia, información e innovación (acorde con las definiciones de Aretio), ha dado un gran paso para asegurar la calidad del sistema.

Por esto, en esta primera capa de este modelo incorporaremos los criterios a tener en cuenta que coinciden con los del modelo de García Aretio, pero refieren a los componentes que hemos descrito en el párrafo anterior:

Metas/Objetivos en relación con Resultados → **Eficacia**
Metas/Objetivos en relación con Recursos, Procesos y Resultados → **Eficiencia**
Necesidades/Demandas del Contexto en relación con Metas/Objetivos→ **Funcionalidad**
Evaluación Externa/ Interna, Estudiantes en vinculación con Resultados → **Información**
Información en vinculación con Mejoras → **Innovación**

En la segunda capa de este modelo proponemos algunos indicadores que se relacionarán en forma directa con los criterios de la primera capa. Para ello, es necesario que determinemos los procesos, recursos y resultados, que consideramos de interés en un sistema de educación a distancia. En la figura 2, pueden verse los procesos principales que luego detallaremos vinculando con recursos y resultados, para acercarnos a los pertinentes de un sistema de EAD.

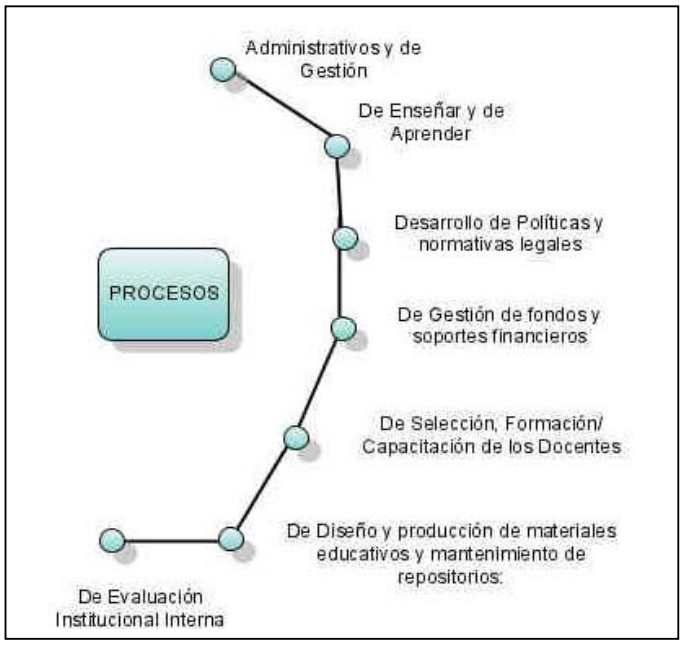


Figura 2. Procesos de principal interés en una IES

4.1.- Indicadores de la capa 2. Los procesos, su vinculación con metas, recursos y resultados

Aquí se mencionan los procesos tenidos en cuenta, su relación con metas, recursos y resultados. Luego se definen como parte de la capa 2, una serie de indicadores que pueden llegar a ayudar en la evaluación de un sistema de EAD.

a) De Administración y Gestión: este proceso incluye establecer circuitos de difusión, inscripción, atención de consultas administrativas, gestión de alumnos y docentes (mantener, registrar y dar información sobre cursos aprobados, notas, certificaciones alcanzadas, etc.). Circuitos para entregar credenciales de acceso a entornos virtuales de enseñanza y aprendizaje (en caso de haberlos), etc.

Algunos Indicadores a considerar:

Vinculados a la eficacia y eficiencia:

- Cantidad de inscripciones (analizando procedencia del alumno) que se reciben, en relación con la cantidad de recursos humanos que atienden dichas inscripciones y circuitos disponibles para la inscripción presencial o a distancia.
- Cantidad de alumnos que inician el proceso luego de la inscripción, en relación con la cantidad de inscriptos. Puede indicar faltas de información administrativa adecuada.
- Cantidad de trámites administrativos iniciados en el día en relación con los finalizados.
- Porcentaje de consultas recibidas a través de los diferentes medios disponibles.

Vinculados a la funcionalidad:

- Cantidad de alumnos y docentes que manifiestan satisfacción respecto de los circuitos administrativos y de consulta disponibles en la institución.

Vinculados a la información e innovación:

- Verificación de que existe un informe que reporte los resultados obtenidos a partir de los indicadores.
- Cantidad de mejoras y cambios implementados en vinculación con el plan de mejoras determinado por la institución.

b) De Enseñar y de Aprender: este proceso incluye, entre otras cosas, planificación y estructuración de los contenidos, definición de objetivos particulares de cada curso, definición de estrategias y actividades didácticas acordes, diseño y desarrollo de materiales instruccionales soportados en diferentes medios atendiendo a diferentes estilos cognitivos (vinculado al proceso de diseño y producción de materiales que se menciona debajo), análisis y definición de metodologías y estrategias de apoyo en TICs, definición de estrategias de interacción y comunicación (análisis de su vinculación con los soportes tecnológicos disponibles, factibilidad de su uso), estrategias de trabajo colaborativo y/o cooperativo, evaluación y redefinición de contenidos, evaluación de medios incorporados, evaluación de los aprendizajes.

Algunos Indicadores a considerar:

Vinculados a la Eficacia

- Cantidad de alumnos que han alcanzado la acreditación en relación con los inscriptos
- Cantidad de alumnos, docentes y coordinadores que han dado una opinión satisfactoria acerca del proceso de enseñanza y de aprendizaje. Esto puede refinarse a distintos aspectos del proceso, como por ejemplo, los referidos a las estrategias de individualización de la enseñanza, interacción, interactividad, mejoramiento de la integración, pensamiento crítico, etc.
- Cantidad de alumnos que presentan un desempeño satisfactorio en cursos posteriores vinculados al actual.
- Cantidad de opiniones satisfactorias (desde determinados ámbitos laborales) respecto del desempeño de los graduados de la institución, en campos vinculados a los contenidos y competencias impartidas.

Vinculados a la Eficiencia

- Cantidad de alumnos asignados por tutor
- Cantidad de consultas recibidas en el proceso de enseñanza en relación con las respondidas
- Cantidad de recursos tecnológicos utilizados como parte de la estrategia didáctica respectos de los disponibles
- Cantidad de actividades didácticas propuestas en relación con la duración del curso y tiempos disponibles.

Vinculados a la funcionalidad

- Expectativas de aprendizaje por parte del alumno respecto de las metas/objetivos de la propuesta
- Expectativas de enseñanza por parte de docentes involucrados respecto de las metas/objetivos propuestos
- Cantidad de graduados insertados en el mercado laboral respecto del total de graduados. Creemos que este indicador puede marcar que las metas del sistema estén acordes a las necesidades del ámbito laboral.

Vinculados a la información e innovación

- Verificación de que existe un informe que reporte los resultados obtenidos a partir de los indicadores.
- Cantidad de innovaciones implementadas de acuerdo a los resultados obtenidos a partir de la evaluación planificada.

c) De Desarrollo de políticas y normativas legales: definición de criterios metodológicos mínimos, definiciones sobre temas como propiedad intelectual,

licencias de software, recomendaciones institucionales, criterios para la selección de docentes y recursos, etc.

Algunos Indicadores a considerar:
Vinculados a la eficacia y eficiencia

- Cantidad de excepciones que se deben realizar a las normas y políticas establecidas
- Cantidad de criterios, y normas que son realmente implementados por los recursos humanos de la institución versus cantidad de criterios y normas establecidos en cada caso.

Vinculados a la funcionalidad

- Cantidad de circunstancias o sucesos ocurridos no contemplados en las reglas y normativas establecidas.

Vinculados a la información e innovación

- Verificación de que existe un informe que reporte los resultados obtenidos a partir de los indicadores.
- Cantidad de innovaciones implementadas de acuerdo a los resultados obtenidos a partir de la evaluación planificada.

d) De Gestión de fondos y soporte financiero/ económico: circuitos para conseguir fondos, distribución equilibrada de los fondos y recursos acorde a las prioridades en las metas/objetivos. Análisis de los soportes económicos para asumir costos de los recursos involucrados, etc.

Algunos Indicadores a considerar:
Vinculados a la funcionalidad

- Cantidad de organismos estatales y privados que financian, motivan, premian, otorgan becas, incentivos para que las IES dispongan de programas para la educación a distancia.

Vinculados a la eficacia y eficiencia

- Cantidad de recursos propuestos en relación con los recursos económicos asignados.
- Cantidad de necesidades/recursos resignados en función de los recursos económicos.
- Metas alcanzadas en relación con costos invertidos
- Comparación del costo invertido por el alumno en un sistema presencial vs. en el sistema a distancia que se está analizando.
- Comparación del costo invertido por la institución en un sistema presencial vs. en el sistema a distancia analizado.

Vinculados a la información e innovación

- Verificación de que existe un informe que reporte los resultados obtenidos a partir de los indicadores.
- Cantidad de innovaciones implementadas de acuerdo a los resultados obtenidos a partir de la evaluación planificada.

e) De Selección, Formación y Capacitación de docentes: determinación de las estrategias y criterios de selección de docentes, definición de los roles tutoriales (administrativos, tecnológicos, académicos, evaluadores, y todos aquellos que la institución considere adecuados), capacitación y formación tutorial, detección de áreas de necesidad de formación, etc.

Algunos Indicadores a considerar:
Vinculados a la funcionalidad

- Expectativas del estudiante respecto de la acción tutorial, en relación con la formación y capacitación dada por la institución a los tutores.
- Cantidad de tutores que ingresan a la institución con formación previa en relación con los requisitos de formación que se exigen desde la institución.

Vinculados a la eficacia y eficiencia

- Cantidad de tutores capacitados/formados en relación con la cantidad de tutores requeridos.
- Cantidad de recursos (tecnológicos, humanos, etc.) a disposición para la capacitación/ formación de los tutores en relación con los tutores a formar. Este indicador debe refinarse teniendo en cuenta los recursos necesarios en cada caso.

Vinculados a la información e innovación

- Verificación de que existe un informe que reporte los resultados obtenidos a partir de los indicadores.
- Cantidad de innovaciones implementadas de acuerdo a los resultados obtenidos a partir de la evaluación planificada.

f) De Diseño y producción de materiales educativos y mantenimiento de repositorios: definición metodológica y roles involucrados, estrategias de diseño, implementación, almacenamiento y disponibilidad (inclusión de repositorios) de los materiales. Vinculación con políticas de propiedad intelectual.

Algunos Indicadores a considerar:
Vinculados a la funcionalidad

- Cantidad de temáticas a tratar en relación con la cantidad de temáticas abordadas en los materiales educativos.
- Disponibilidad de entornos flexibles que soporten el depósito y/o publicación de materiales y documentos de apoyo para el aprendizaje de los contenidos del programa.
- Disponibilidad de formatos de materiales variados en función de las necesidades de los alumnos y docentes.
- Cantidad de materiales que respetan estándares de accesibilidad.
- Cantidad de materiales que respetan estándares de e-learning.

Vinculados a la eficacia y eficiencia

- Cantidad de recursos (diferentes expertos, herramientas tecnológicas, etc.) asignados en función de los requerimientos de diseño y producción de materiales.
- Opinión de alumnos y docentes respecto de la claridad en la presentación de los contenidos.
- Opinión de alumnos, docentes y coordinadores respecto de si los contenidos de los materiales están acordes a los objetivos del curso o programa.
- Opinión de alumnos, docentes y coordinadores respecto de si los contenidos son pertinentes y actuales.
- Opinión de alumnos, docentes y coordinadores respecto de si los contenidos favorecen el aprendizaje, la reflexión crítica, el análisis y la investigación.

Vinculados a la información e innovación

- Verificación de que existe un informe que reporte los resultados obtenidos a partir de los indicadores y planes de evaluación realizados (referimos aquí a los que son específicos de los materiales involucrados en los cursos).
- Cantidad de innovaciones implementadas de acuerdo a los resultados obtenidos a partir de la evaluación planificada.

g) De Evaluación interna: definición de objetos de evaluación, definición del plan de evaluación (teniendo en cuenta momentos, instrumentos, actores involucrados, etc.), generación de resultados, y planes de mejora.

Algunos Indicadores a considerar:
Vinculados a la funcionalidad

- Grado en que la evaluación interna se apega a estándares de evaluaciones nacionales e internacionales.

Vinculados a la eficacia y eficiencia

- Cantidad de recursos asignados en relación a las necesidades de los procesos de evaluación. Este punto debe refinarse de acuerdo a la detección de necesidades en cada caso.
- Opinión por parte de docentes, alumnos y directivos acerca de la calidad de los instrumentos de evaluación desarrollados
- Cantidad de aspectos sugeridos, por los recursos humanos, involucrados en la evaluación que no han sido tomados en cuenta por los instrumentos.

Vinculados a la información e innovación

- Verificación de que existe un informe que reporte los resultados obtenidos a partir del plan de evaluación interna.
- Cantidad de innovaciones implementadas el plan de evaluación interna atendiendo a las sugerencias no tenidas en cuenta previamente.

Hasta aquí hemos considerado los criterios generales, luego determinamos algunos indicadores a nivel de procesos, de manera tal de poder evaluar la calidad de un sistema de educación a distancia.

A continuación describiremos, sin entrar en detalles, la tercera capa de nuestro modelo. Esta capa deberá tener en cuenta cómo establecer una medida para los indicadores sugeridos.

4.3.- La capa 3. Medidas a considerar para los indicadores

Para no hacer extenso este trabajo, tomaremos como ejemplo uno de los procesos con sus indicadores, y explicaremos cómo debería desarrollarse la capa 3.

Supongamos que referimos a los procesos de administración y gestión, al llegar a la capa 3, la institución deberá determinar cuál sería la tasa que indica que no hay recursos humanos y/o circuitos administrativos por debajo (o por encima) de lo necesario en relación con la cantidad de inscripciones (Indicador: Cantidad de inscripciones que se reciben, en relación con la cantidad de recursos humanos que atienden dichas inscripciones y circuitos disponibles para la inscripción presencial o a distancia).

Plantearemos un caso, pensemos en un programa de educación a distancia de una facultad (nivel postgrado), donde se permite la inscripción vía Internet y/o presencialmente, y existen 3 personas que atienden las inscripciones presenciales, y una los formularios recibidos desde Internet. La cantidad de inscriptos vía Internet es 1000, y la cantidad de inscriptos presencialmente es 100, evidentemente, los recursos humanos están mal distribuidos.

Así, la IES deberá considerar cada uno de los indicadores abordados y establecer las medidas en forma contextualizada.

Esta capa podría explotarse considerando una institución específica y definiendo las medidas en base a recursos, estrategias y particularidades del contexto. Es por ello que solo se ha planteado un ejemplo para mostrar a qué referimos con las medidas de la capa 3.

Actualmente se está completando el desarrollo de este modelo para un caso de estudio en el marco de la UNLP de manera tal de encontrar sus debilidades y proponer futuras mejoras.

5.- Conclusiones

Se ha realizado un estudio e investigación referido al tema de la Calidad en Educación Superior y en particular en la Educación a Distancia, presentando diferentes opiniones y los debates actuales en la temática.

Se presentó una propuesta de modelo de evaluación de la calidad desarrollado en capas donde en la primera de ellas se establece criterios generales tomados de modelos de calidad total.

En la segunda capa, con el propósito de acercarnos más a las especificidades del ámbito académico y en particular de la Educación a Distancia, se sugirieron indicadores vinculados con los criterios generales. Estos indicadores permiten poner el foco en los procesos de enseñar y aprender y en las cuestiones administrativas, organizacionales y de infraestructura tecnológica requeridas en la modalidad EAD.

La tercera capa tiene el propósito de especificar las medidas que permitirán evaluar si los indicadores alcanzan la calidad esperada por la institución.

Como se mencionó anteriormente, en la actualidad se está trabajando en la aplicación del modelo en un caso específico, en el que participará una institución con las variables propias de la misma.

6.- Referencias bibliográficas

[1] Declaración Mundial sobre la Educción Superior en el siglo XXI: visión y acción. http://www.unesco.org/education/educprog/wche/declaration_spa.htm#declaracion
[2] “De la visión a la acción”. Artículo 11. Evaluación de la calidad. Declaración Mundial sobre la Educción Superior en el siglo XXI: visión y acción.
http://www.unesco.org/education/educprog/wche/declaration_spa.htm#declaracion
[3] Barberá, E. (coord.), Badía, A., Mominó, J.“La incógnita de la Educación a Distancia”. ICE – Universidad de Barcelona – Horsori. Barcelona. 2001.
[4] http://www.tqm.es/TQM/ModEur/ModeloEuropeo.htm
[5] Sangrà Albert. “La calidad en las experiencias virtuales de educación superior”. Director académico de la UOC. 2001.
[6] Villanueva E. Ex presidente de la Red Iberoamericana para la Acreditación de la Calidad de la Educación Superior (RIACES). Entrevista en la Revista Cognición (2006). ISSN: 1850-1958
[7] Moore, Kearsley. “Distance Education. A System View” Capítulo 9. Wadsworth Publishing Company. 1996.
[8] García Aretio, L. (coord.), Corbella M., Dominguez Figaredo D. “De la Educación a Distancia a la Educación Virtual”. Editorial Ariel. 1er Edición. ISBN:978-84-344-2666-5. 2007.
[9] Zamora G. Pamela. “Buenas Prácticas y Criterios de Calidad en la Enseñanza Superior a Distancia”. http://www.puc.cl/citeduc/web/html/sub_articulos.htm.
[10] De Giusti, Gorga, Madoz. Una propuesta para evaluar la calidad de Sistemas de Educación no presenciales basados en la web. GCETE 2005. Brasil.
[11] Gorga, Madoz, Feierherd, Depetris. Una propuesta de métrica para evaluar sistemas de Educación a Distancia basados en Internet. CACIC2002. Argentina.

El uso del foro en el Aprendizaje colaborativo: una experiencia de la cátedra Tecnología Educativa II de la Licenciatura en Tecnología Educativa UTN FRRe.

Lic. María del Carmen Maurel
Licenciatura en Tecnología Educativa – Grupo de Investigación Educativa
Universidad Tecnológica Nacional – Facultad Regional Resistencia
Resistencia, Chaco 3500. Argentina
mmaurel@frre.utn.edu.ar – mmaurel_38@yahoo.com.ar

y

Mg. Liliana Raquel Cuenca Pletsch

Licenciatura en Tecnología Educativa – Grupo de Investigación Educativa
Universidad Tecnológica Nacional – Facultad Regional Resistencia
Resistencia, Chaco 3500. Argentina
cplr@frre.utn.edu.ar – cplr@arnet.com.ar

Abstract

In the last five years a group of teachers and researchers of the Educational staff of the Resistencia Regional Faculty of the UTN has developed some activities to improve the learning process. The students of the Educative Technology course and those who attend the Introductory Seminary for the Information Systems Engineering were included in these activities.

The students'profiles of both careers is very different. Whereas the students of the Educational Technology are teachers with a degree in Education or with a technical degree from the UTN, those who attend the Introductory Seminary are teenagers.

For the students of the Educational Technology, the goal was learn how to work in virtual environments through a real experience, focusing in collaboration. The results of this experience which started in 2005 are presented in this work.

Pedagogical and technological concepts are, in first place, established for this work by the teachers on the basis of "collaborative learning" and its effects, specially focused in how to learn, the acceptation of their companions and the academic performance. In second place, it refers to the characteristics of the "collaborative learning" exposed by the students in this experience, and finally as a conclusion, a summary of the strengths and weaknesses of the developed experience is shared.

Keywords: collaborative learning, virtual environments, virtual forum.

Resumen

En los últimos cinco años un grupo de docentes e investigadores en el área educativa de la Facultad Regional Resistencia de la UTN ha desarrollado actividades tendientes a mejorar el aprendizaje de los alumnos de la Licenciatura en Tecnología Educativa y de aquellos que cursan el Seminario de Ingreso a la carrera de Ingeniería en Sistemas de Información bajo las modalidades a distancia o semipresencial.

El perfil de los alumnos de ambas ofertas es muy diferente. En tanto que los alumnos de la Licenciatura son docentes con títulos terciarios o bien egresados con títulos intermedios de la UTN, quienes cursan el Seminario de ingreso son, en su mayoría, adolescentes.

En este trabajo se presenta la experiencia iniciada en el año 2005 en asignaturas del área Tecnología Educativa de la Licenciatura para que nuestros alumnos aprendan, mediante la experiencia, a trabajar colaborativamente en entornos virtuales.

En primer lugar y sobre las bases conceptuales del aprendizaje colaborativo y sus efectos, especialmente en lo referido a la forma de aprender, a la aceptación de los otros y al rendimiento académico, se exponen las definiciones pedagógica y técnica determinadas por los docentes para la experiencia. En segundo lugar se refiere a las características del aprendizaje colaborativo expuestas por nuestros alumnos en esta experiencia; y a manera de cierre se comparte un resumen de las fortalezas y debilidades de la experiencia desarrollada.

Palabras clave: aprendizaje colaborativo, entornos virtuales, foros virtuales.

1. BASES CONCEPTUALES DEL APRENDIZAJE COLABORATIVO

Al comparar el modelo de aprendizaje colaborativo con otros modelos de aprendizaje, por ejemplo el modelo de presentación o el modelo de enseñanza de conceptos, encontramos, una diferencia fundamental. El modelo de aprendizaje colaborativo trasciende la problemática académica de adquirir información, procesarla y adquirir e incorporar nuevas destrezas y conocimientos. El modelo que ahora estudiamos presenta importantes objetivos sociales. Relaciona la actual vida social del alumno con la vida y el trabajo futuro, donde le van a ser necesarias las destrezas sociales que propone el modelo.

El punto de partida de este modelo no se puede atribuir a un autor o a una sola corriente pedagógica. Algunos autores se remontan a la pedagogía griega en busca de los orígenes. Otros autores comienzan la fundamentación teórica en las corrientes psicopedagógicas de principios del siglo XX (Arends, 1994).

Las orientaciones pedagógicas de John Dewey, sobre todo desde la publicación de su libro Democracy and Education (1916), insistían en que el aula debía ser un laboratorio de la vida real, el docente debía crear un ambiente de aprendizaje caracterizado por los procedimientos democráticos y por los procesos científicos, los alumnos debía aprender y resolver problemas trabajando en pequeños grupos y en permanente interacción.

Años más tarde Herbeit Thelen desarrolló de forma más precisa las peculiaridades de trabajo en grupo en el aula, facilitando las bases para lo que hoy llamamos aprendizaje cooperativo. El planteamiento tanto de Dewey como de Thelen sobrepasaba el mero aprendizaje académico para fundamentar, desde la escuela, los comportamientos humanos en las sociedades democráticas.

Decíamos al principio que el aprendizaje colaborativo podía llevarse a cabo a distancia, mediante el uso de las Tecnologías de la Información y las Comunicaciones (TIC), promoviendo el trabajo conjunto entre distintos centros docentes, de países y razas diferentes. Un análisis de la Historia de la Educación nos muestra que la insistencia en el trabajo en grupo y en la cooperación entre los alumnos se reforzó cuando, tras el decreto sobre la integración racial en las Escuelas de los EE.UU. (1954), Gordon Allport propuso una serie de medidas pedagógicas encaminadas a conseguir una verdadera “integración” escolar a base de trabajo en grupo y proyectos comunes. En este mismo

clima de escuelas con problemas de integración se desarrolla el trabajo de uno de los autores más importantes para la fundamentación del aprendizaje colaborativo Robert Slavin (1983, 1984, 1990, 1993).

Los estudios de los expertos en psicología social demostraban que, frecuentemente, había diferencias notables entre el desempeño de tareas físicas o intelectuales individualmente o en presencia de otras personas. Algunos pedagogos aprovecharon estas investigaciones para orientar sus estrategias docentes de trabajo en grupo.

Uno de los objetivos más importantes de la escolarización es la socialización de los alumnos y su integración en la sociedad. Si deseamos una sociedad democrática y participativa tendremos que enseñar a los estudiantes a ser democráticos y participativos.

El proceso de socialización es largo y complejo, y en un mundo como el que nos ha tocado vivir, debe estar abierto a la comprensión de todas las culturas. Desde que el niño se incorpora a la primera jornada de clase en Educación Infantil comienza su aprendizaje en la cooperación. Primero aprende que no está solo, tiene otro niño u otra niña que está al lado y que luego puede jugar con él. Hay un material didáctico común que todos pueden usar y que deben respetar. La noción de interacción e interdependencia comienza a tener fuerza y a crecer.

Los pedagogos destacan muchas ventajas en el trabajo en grupo. Veamos por ejemplo algunas ventajas de las que cita Fabra (1994:27):

- Favorecen el hecho de que alumnos y alumnas aprendan a aceptarse mutuamente y a cooperar.
- Logran que se sientan miembros de un grupo y experimenten hacia ese grupo gran variedad de sentimientos positivos, a la vez que aumentan su seguridad personal al sentirse apoyados por él.
- Estimulan los diversos aprendizajes y favorecen la actividad dirigida hacia el logro de unos objetivos previamente establecidos.
- Facilitan el desarrollo intelectual y afectivo del estudiantado, que puede realizar diversos ejercicios y actividades que constituyen una base sólida de experimentación sobre la cual puede construir su visión del mundo y del desarrollo de la ciencia.
- Mejoran el clima o atmósfera de la clase en el sentido de que disminuyen tensiones y malos humores y consiguen un ambiente suficientemente relajado y entusiasta como para que tanto estudiantes como docentes puedan trabajar a gusto y con eficacia.

2- DESCRIPCIÓN DE LA PROPUESTA

La Licenciatura en Tecnología Educativa de la UTN es un ciclo de licenciatura que establece como requisito para ingresar el tener un título terciario docente de 4 años, o bien un título terciario de 4 años e importantes antecedentes docentes o bien título intermedio de la UTN.

Los alumnos de la carrera son docentes de nivel medio y superior universitario y nuestro desafío es lograr disminuir la brecha entre los estudiantes de esos dos niveles, a quienes Marc Prensky (2001) denomina *"nativos digitales"*, y sus profesores (nuestros alumnos), a quienes el mismo autor denomina *"inmigrantes digitales"*. Según estas definiciones el *nativo digital* se distingue por: recibir información de manera sumamente rápida, preferir los procesos paralelos, preferir el gráfico antes que el texto, defender los accesos al azar a partir de hipertextos, funcionar mejor cuando trabajan en red, prosperar con satisfacción inmediata y bajo recompensas frecuentes y elegir "jugar en serio"

antes que trabajar. Los *inmigrantes digitales* no valoran estas habilidades y eligen enseñar lentamente, paso a paso, una cosa a la vez, individualmente. No entienden la poca paciencia de sus alumnos para las clases magistrales y la lógica paso a paso. Así las cosas, los nativos digitales terminan la escuela y el nivel Superior de la forma tradicional, sin que sus docentes logren hacer significativos sus aprendizajes.

El desafío para los educadores es aprender a comunicarse en la lengua y el estilo de sus estudiantes, lo cual no significa cambiar el significado de lo importante o pensar en otras habilidades, sino "ir más rápido, profundizar en paralelo, acceder a la información siguiendo el azar". No significa dejar de enseñar la lectura, la escritura, las matemáticas, el pensamiento lógico, sino enfocarlo a una era distinta. Y éste es el desafío de las materias del área Tecnología Educativa: ayudar al "inmigrante digital" a entender esta nueva forma de aprender y "pensar en nuevas maneras de enseñar las viejas materias"

En el contexto del desarrollo de la cátedra Tecnología Educativa II se trabaja como eje temático el aprendizaje colaborativo. Dado que el sustento pedagógico de la cátedra es que todo saber se aprende mejor haciendo, nos proponemos que los estudiantes vivencien el aprendizaje colaborativo virtual, que es similar al presencial en cuanto a su concepción pero diferente en cuanto a su implementación. Como todo aprendizaje debe ser, además, significativo, decidimos que la vivencia se realizara en el marco de elaboración del Trabajo Final de la cátedra.

Los objetivos de la propuesta son:

- Vivenciar de manera práctica el aprendizaje colaborativo virtual.
- Identificar sus fortalezas y debilidades..
- Comprender que el trabajo colaborativo virtual, al igual que el presencial, es un todo construido desde el conjunto que conforma el grupo.

El trabajo colaborativo necesita de la cooperación, entendida ésta como un proceso social de construcción. Partiendo de esta base, podemos entender este aprendizaje como un proceso en el que cada individuo aprende más de lo que aprendería por sí mismo, debido a la interacción con otros miembros de su grupo. Desde esta perspectiva, el resultado de un trabajo hecho en un grupo tiene un valor superior al que tendría la suma de los trabajos individuales de cada miembro de dicho grupo.

3. DEFINICIÓN Y ESPECIFICACIONES PEDAGÓGICAS Y TÉCNICAS DE LA PROPUESTA

Durante el primer semestre del año 2005 se definieron las especificaciones pedagógicas y técnicas que han sido el punto de partida para poner en práctica esta experiencia piloto de aprendizaje colaborativo iniciada con la cohorte 2005.

En cuanto a las especificaciones pedagógicas, se definieron las estrategias que podrían favorecer el trabajo colaborativo, y que son:

- El conocimiento previo de los miembros que participan en el grupo y su contexto.
- La finalidad de la propuesta debería contemplar que los estudiantes trabajen para resolver un problema conjunto.

- La planificación de tareas de trabajo individual previas relacionadas con la investigación y análisis de la información relativa a la propuesta, para poder aportar elementos de discusión posterior.
- Una planificación detallada del trabajo a realizar, especificando las diferentes fases y negociando con los diferentes integrantes del equipo las posibilidades de que éste se cumpla.
- Una interacción periódica en el proceso de realización de la propuesta.
- Un trabajo final que incluya la realización de un producto común como respuesta a los objetivos definidos previamente.
- Una evaluación que tenga presente el proceso y los resultados por parte de todos los participantes.

Se determinó también que sería importante considerar otras estrategias relativas a las actitudes y valores que deberían compartir los participantes en la propuesta de aprendizaje, y que son:

- Lograr que todos los miembros compartan toda la información y que tengan en cuenta las sugerencias de los compañeros que aporten nuevos elementos a la reflexión.
- Fomentar la argumentación y la búsqueda de razones que apoyen las afirmaciones, sugerencias y opiniones.
- Valorar la capacidad de llegar a un consenso cuando es necesario realizar una acción.
- Promover la aceptación de que el responsable de la toma de decisiones es el grupo, y no un solo miembro.
- Valorar la necesidad de compartir una temporización y compromiso establecidos.

Desde una perspectiva más funcional, se precisaron las características que debería tener la herramienta informática a utilizar:

- Ser fácil e intuitiva
- Tener una interfase clara
- Ser una herramienta accesible vía Internet
- Responder a las necesidades pedagógicas de los estudiantes, mencionadas más arriba.

En función a esta última característica, se describieron también las funcionalidades y las especificaciones técnicas que debería tener la herramienta informática utilizada para el trabajo colaborativo. De forma general, estas funcionalidades se podrían agrupar en las siguientes categorías:

- Herramienta de diálogo asíncrono: ofrecer las funcionalidades necesarias para llevar a cabo discusiones, diálogos, toma de decisiones, aspectos de suma importancia para desarrollar cualquier trabajo colaborativo.
- Herramienta de archivo*:* que permita compartir diferentes tipos de archivo: documentos de texto, imagen, sonido, que deban ser manipulados por más de un miembro del equipo.
- Registro de la actividad colaborativa esta herramienta permite realizar un control y seguimientos de los trabajos desarrollados por los estudiantes y de la frecuencia y calidad de su participación.

Todas estas especificaciones están cubiertas por la herramienta de foro interno del campus virtual, Moodle, adaptado para su utilización en la Licenciatura.

4. DESCRIPCIÓN DE LA EXPERIENCIA

Como se señalara más arriba la propuesta consiste en una experiencia de aprendizaje colaborativo para los alumnos de la cátedra Tecnología educativa II. En la misma, como tarea del eje temático relacionado con el aprendizaje colaborativo, se inscribe a los alumnos en un foro denominado Trabajo Final. Este foro tiene como objetivo elaborar colaborativamente el trabajo final de la cátedra.

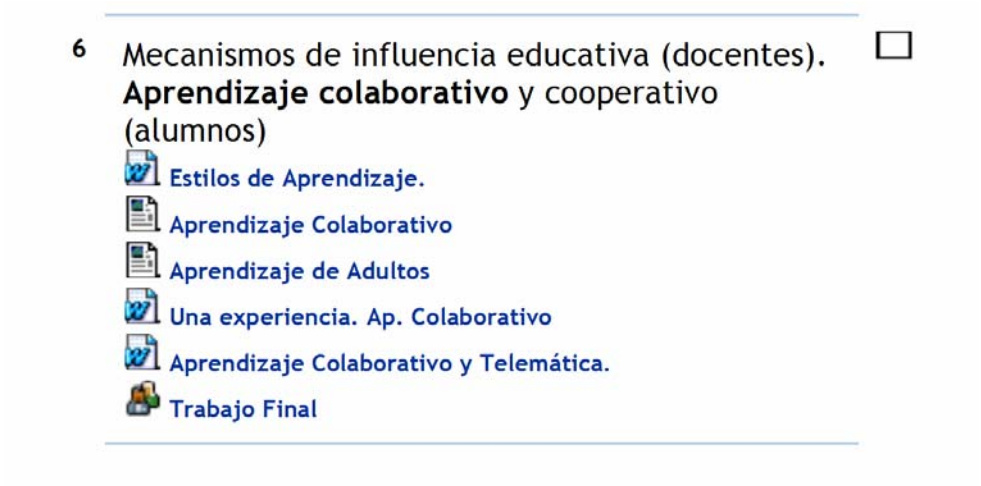


Fig. 1. Pantalla de la unidad que trata sobre el trabajo colaborativo

En los años 2005 y 2006 se trabajó en el diseño de un portal educativo. Actualmente se trabaja en la implementación del mismo.

Para el presente ciclo lectivo se propuso a los alumnos elaborar un proyecto formativo a distancia. El tema se eligió en forma conjunta entre la cátedra y los alumnos. La consigna fue que se tratara de una propuesta tendiente a resolver alguno de los principales problemas de formación de los alumnos que terminan el nivel medio e ingresan al nivel superior. Se descartó el tema *Metodología de estudio* debido a que los alumnos de la cohorte 2006 lo habían desarrollado e implementado a distancia para aprobar la asignatura Tecnología Educativa III. Luego de un interesante debate e intercambio de ideas se eligió como tema el *aprendizaje basado en problemas* destinado a estudiantes de EGB3.

La primera etapa de consensos, sobre la temática a desarrollar el proyecto formativo y los destinatarios, es siempre la más difícil y demanda alrededor de dos o tres semanas. Esto es así porque a la dificultad de ponerse de acuerdo sobre un tema determinado se le suma la necesidad de hacerlo en forma virtual. En esta instancia es fundamental la presencia de los docentes incentivando la participación e iniciando las discusiones. Es también en este momento del trabajo cuando los alumnos perciben la importancia de respetar y analizar todos los puntos de vista, ordenar las discusiones y pautar el momento del consenso.

En una segunda etapa se discute la organización y distribución de tareas, la cual deberá ser entendida no como suma de partes o individualidades, sino como una cuestión de organización en la que para cada tarea se definen responsables pero en cuya ejecución deben intervenir todos.

Una vez definidas las responsabilidades, en algunos casos individuales y en otros grupales, se establece el cronograma de trabajo. Cada alumno o grupo responsable de una tarea debe abocarse a la

misma y mantener informado al resto sobre los avances, a la vez que deberá solicitar opiniones y sugerencias en los momentos en que fuera necesario. Por cada tarea se abre un tema en el foro a los efectos de mantenerlo ordenado y que los alumnos y docentes no pierdan el hilo argumental.

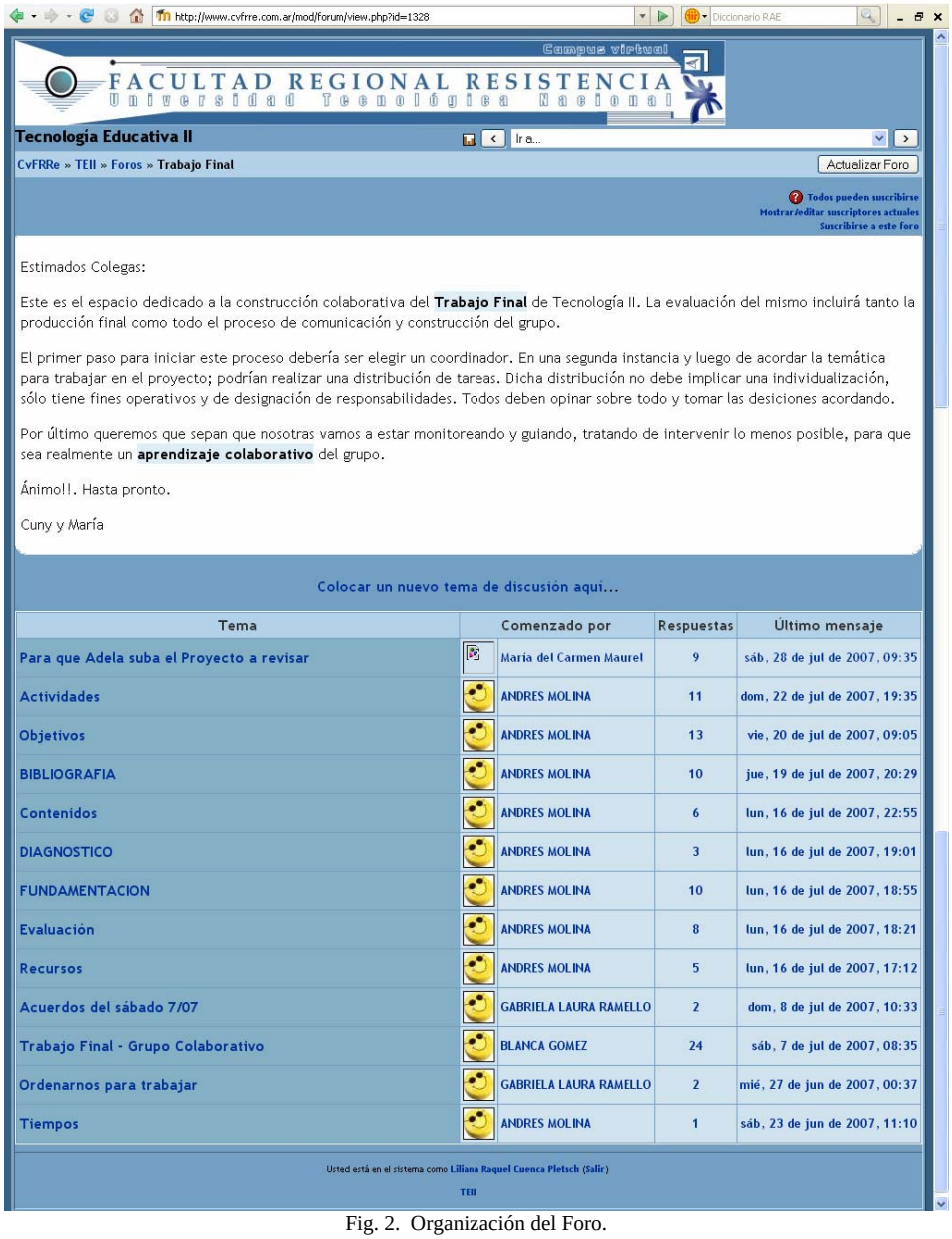
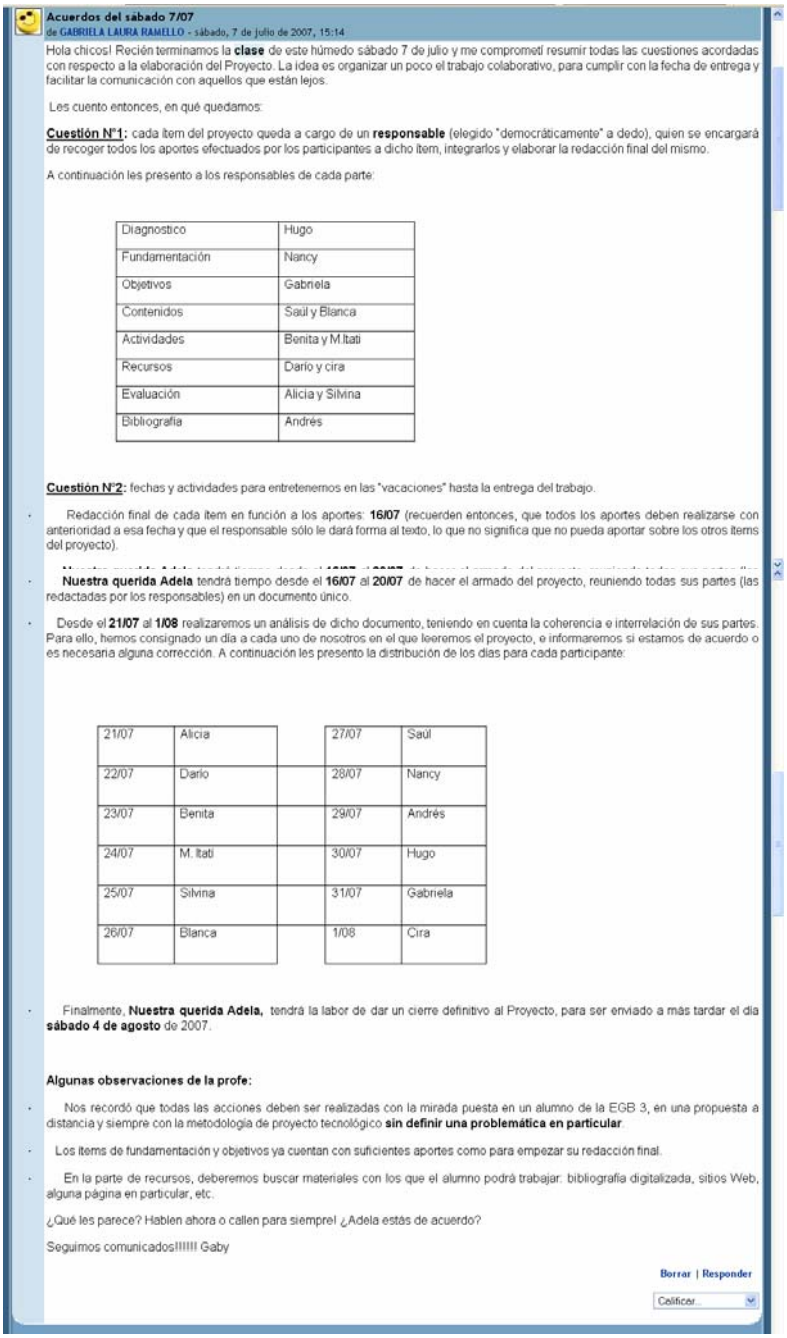


Fig. 2. Organización del Foro.

Dado que lo asistencia a los encuentros presenciales no es obligatoria, algunos alumnos sólo asisten a la facultad los días en que deben rendir los exámenes y realizan todo el cursado y las actividades completamente a distancia. Es así que, para este trabajo muchas veces sucede que los alumnos que asisten a estos encuentros logran acuerdos y consensos que luego deben publicar en el foro para dar oportunidad a todos de opinar y decidir. En este sentido, con la guía y orientación permanente de los docentes se ha establecido una comunicación enriquecedora y constructiva, de forma que todos

han participado y quienes no concurren habitualmente a los encuentros presenciales, pudieron aportar, elaborar, opinar, disentir.



Acuerdos del sábado 7/07
de GABRIELA LAURA RAMELLO - sábado, 7 de julio de 2007, 15:14

Hola chicos! Recién terminamos la **clase** de este húmedo sábado 7 de julio y me comprometí resumir todas las cuestiones acordadas con respecto a la elaboración del Proyecto. La idea es organizar un poco el trabajo colaborativo, para cumplir con la fecha de entrega y facilitar la comunicación con aquellos que están lejos.

Les cuento entonces, en qué quedamos:

Cuestión N°1: cada ítem del proyecto queda a cargo de un **responsable** (elegido "democráticamente" a dedo), quien se encargará de recoger todos los aportes efectuados por los participantes a dicho ítem, integrarlos y elaborar la redacción final del mismo.

A continuación les presento a los responsables de cada parte:

Diagnostico	Hugo
Fundamentación	Nancy
Objetivos	Gabriela
Contenidos	Saúl y Blanca
Actividades	Benita y M.Itati
Recursos	Darío y cira
Evaluación	Alicia y Silvina
Bibliografía	Andrés

Cuestión N°2: fechas y actividades para entretenernos en las "vacaciones" hasta la entrega del trabajo.

- Redacción final de cada ítem en función a los aportes: **16/07** (recuerden entonces, que todos los aportes deben realizarse con anterioridad a esa fecha y que el responsable sólo le dará forma al texto, lo que no significa que no pueda aportar sobre los otros ítems del proyecto).
- **Nuestra querida Adela** tendrá tiempo desde el **16/07** al **20/07** de hacer el armado del proyecto, reuniendo todas sus partes (las redactadas por los responsables) en un documento único.
- Desde el **21/07** al **1/08** realizaremos un análisis de dicho documento, teniendo en cuenta la coherencia e interrelación de sus partes. Para ello, hemos consignado un día a cada uno de nosotros en el que leeremos el proyecto, e informaremos si estamos de acuerdo o es necesaria alguna corrección. A continuación les presento la distribución de los días para cada participante:

21/07	Alicia		27/07	Saúl
22/07	Darío		28/07	Nancy
23/07	Benita		29/07	Andrés
24/07	M. Itatí		30/07	Hugo
25/07	Silvina		31/07	Gabriela
26/07	Blanca		1/08	Cira

- Finalmente, **Nuestra querida Adela,** tendrá la labor de dar un cierre definitivo al Proyecto, para ser enviado a más tardar el día **sábado 4 de agosto** de 2007.

Algunas observaciones de la profe:

- Nos recordó que todas las acciones deben ser realizadas con la mirada puesta en un alumno de la EGB 3, en una propuesta a distancia y siempre con la metodología de proyecto tecnológico **sin definir una problemática en particular**.
- Los ítems de fundamentación y objetivos ya cuentan con suficientes aportes como para empezar su redacción final.
- En la parte de recursos, deberemos buscar materiales con los que el alumno podrá trabajar: bibliografía digitalizada, sitios Web, alguna página en particular, etc.

¿Qué les parece? Hablen ahora o callen para siempre! ¿Adela estás de acuerdo?

Seguimos comunicados!!!!!! Gaby

Borrar | Responder

Calificar...

Fig. 3. Comunicación de consensos logrados en un encuentro presencial

Para finalizar el proyecto los alumnos dedican entre dos y tres semanas para que todos los participantes puedan realizar una revisión final. Se establecen fechas tope para enviar las observaciones, para acordar sobre las mismas y para que quien oficie de redactor realice las modificaciones que correspondan.

La elección de un redactor es fundamental para asegurar la coherencia en la presentación del proyecto. Este rol no implica que quien lo ejerce tendrá una actitud pasiva durante la elaboración del mismo.

Una vez que el proyecto está terminado deben habilitarse dos nuevos temas en el foro: uno denominado *Proyecto*, donde los alumnos deben subir el archivo que contiene el trabajo requerido por la cátedra y otro denominado *Evaluación* donde los alumnos deben volcar sus apreciaciones sobre la experiencia llevada a cabo indicando: problemas que debieron sortear, habilidades que pudieron desarrollar, cuál fue el aporte de sus compañeros, de la asignatura y de los docentes, y cómo creen que esta experiencia aportará a su desempeño futuro.

5. RESULTADOS

Habiendo analizado la experiencia en los años anteriores, y en un ejercicio de anticipación respecto de la actual, se plantean algunos aspectos significativos extraídos del proceso de trabajo colaborativo virtual realizado hasta ahora, aspectos que se ven confirmados por las valoraciones hechas por los estudiantes. Así pues, esta experiencia:

- *Desarrolla habilidades de colaboración y trabajo en equipo:*

  "...Creemos que es un buen sistema de trabajo y, después de la experiencia, será más fácil afrontar otro. El hecho de conocer el funcionamiento de un aprendizaje en grupo virtual facilitará un trabajo posterior".

- *Amplía los recursos de los cuales dispone el estudiante*

  "… A muchos nos obligó a pasar más horas de las que hubiésemos querido navegando en Internet, pero nos ayudó a mejorar nuestras habilidades para la búsqueda de información, el acceso a bases de datos, a bibliotecas virtuales y a portales. Sin querer aprendimos que las herramientas de chat sirven también para trabajar y que los foros pueden ser organizados para facilitar el aprendizaje. Ahora podemos enseñar a nuestros hijos y alumnos a ser usuarios inteligentes de Internet, antes pensábamos que navegar era una pérdida de tiempo"

- *Estimula el aprendizaje*

  "... El grupo facilita, se encuentra colaboración, nos apoyamos unos a otros, nos exigimos más".

- *Motiva al estudiante para aprender y continuar superándose.*

  "... Trabajar colaborativamente nos posibilita lograr más fácilmente los objetivos propuestos "

  " ….surgen aspectos a seguir trabajando, ideas para mejorar lo que estamos haciendo y en algunos casos puntas para trabajar nuevos proyectos"

- *Contextualiza el estudio individual*

  "...Para todo estudiante es conveniente contar con un grupo de estudio; si bien el trabajo de fijación de los contenidos es personal y cada ser humano tiene su propio ritmo de aprendizaje, el grupo de estudio es un espacio propicio para: confrontar el pensamiento de cada uno y aceptar la disidencia, ser receptor y productor de información, lograr un lugar de pertenencia y de referencia, afirmar la identidad, perder el individualismo y ganar en cooperación."

- *Aporta contraste, crítica y permite incorporar nuevos puntos de vista.*

  "… al analizar una idea por medio escrito y asíncrono, disponemos de más tiempo para reflexionar y elaborar aportes más críticos y profundos que cuando el debate se da en forma oral y en tiempo real"

  "… al disponer de respuestas que se mantienen en el tiempo, es posible analizar mejor los diferentes puntos de vista lo cual no suele suceder en la inmediatez de la oralidad".

- *Amplía las formas de interpretar y analizar la realidad*

  "…a diferencia de cuando debemos producir en forma individual, en esta modalidad estamos obligados a analizar y evaluar las opiniones y aportes de los demás. Esto presenta el inconveniente de que nos retrasa en las decisiones, pero al final terminamos enriquecidos con la experiencia transmitida por personas que se desempeñan en otras ciudades y provincias."

- *Fomenta valores como la solidaridad.*

  "… es importante el apoyo que nos dan nuestros compañeros cuando se dan cuenta de que tenemos alguna dificultad para entender algún tema que se está tratando o cuando nos envían material o enlaces relacionados con las tareas que tenemos asignadas. Muchas veces esas actitudes, más el apoyo e incentivo de los profesores, son los que nos ayudan a seguir."

Algunas de las dificultades detectadas son:

- *El requerimiento de tiempo de dedicación.*

  "... Hemos tenido que dedicar mucho tiempo, lo que ha influido en el tiempo general de estudio de otras asignaturas y también en la realización de las actividades laborales"

- *Las dificultades de adaptación al nuevo entorno*

  "... los estudiantes se resisten a incorporarse a un espacio nuevo, están muy acostumbrados a la presencialidad y al individualismo al que muchas veces los lleva la falta de tiempo. Esto se complica aún más si no son expertos en el manejo de las TIC, si no tienen conexión a Internet en su casa o si la conexión no es la adecuada".

- *Dificultad en la evaluación y reconocimiento del trabajo individual.*

  "...Si no se realiza un seguimiento puntual de la actividad desarrollada continuamente por cada alumno, no se puede determinar el compromiso de participación de cada uno y la calidad de intervención. Este trabajo demanda una dedicación especial por parte de los docentes que van a evaluar el trabajo final".

- *Se alargan y complican los procesos de coordinación y de toma de decisiones.*

  "... Es posible encontrarse con personas que no participan, que no se integran en el ritmo de trabajo, se pierden días esperando la respuesta, se ausentan del grupo sin dar explicaciones".

- *Introduce nuevas dificultades de gestión de la labor docente:*

  "... Mantener la actividad grupal en el aprendizaje virtual es muy difícil, en algunos casos los estudiantes no se conocen, el encuentro virtual presenta dificultades en las personas que no están ambientadas a él. Una crítica de alguien a quien no se ha visto nunca puede producir una disminución en la motivación para seguir adelante con la actividad. En ese momento es fundamental la tarea del docente que gestiona la propuesta; de lo contrario si no hay una atención

permanente a lo que está sucediendo en el grupo virtual, podría suceder que un grupo de cinco o seis alumnos quede reducido a solo tres".

5. CONCLUSIONES

La experiencia desarrollada surge a partir de la preocupación compartida por generar, a partir de los foros incorporados en la mayoría de las plataformas educativas, oportunidades para lograr diferentes tipos de aprendizaje y comunicación. Se planteó una interactividad bidireccional y multidireccional de la cual se pueden extraer algunas conclusiones:

- La propuesta de aprendizaje presentada a los alumnos se ha convertido en una propuesta de autoaprendizaje para los propios docentes involucrados en la experiencia.
- Se ha detectado en cada caso que se debe iniciar la experiencia de acuerdo con los conocimientos previos del grupo.
- Es muy importante la planificación de las actividades y la designación de responsabilidades y cronogramas. Esto es así porque los estudiantes adultos tienen múltiples compromisos laborales y tienden a postergar las tareas que no tienen deadline. Además el sentido de responsabilidad los obliga a cumplir con los cronogramas y las funciones asumidas.
- Es sumamente importante incluir entre los criterios de evaluación las actitudes y valores que se han enunciado en el ítem 3.
- La intervención del docente debe ser oportuna, ordenando el debate, coordinando las tareas, facilitando alternativas, corrigiendo permanentemente. Pero es necesario que dicha intervención se establezca a partir de un liderazgo sensible que no empañe el aprendizaje del grupo. El grupo debe construir por si mismo.
- Por último cabe reconocer que, como en toda experiencia educativa, se puede observar que el papel fundamental del docente es construir con márgenes y buscar el punto medio para que los alumnos puedan ser autogestionarios de sus propios aprendizajes.

Quedan muchos aspectos por analizar a partir de esta experiencia, uno de ellos es indagar si esta metodología de aprendizaje del trabajo colaborativo virtual es aplicable a cualquier perfil de alumno o debe pensarse en modelos diferentes para perfiles diferentes. Este tema será abordado en proyectos futuros.

Bibliografía

1. Alonso,C.M.; Gallego, D.J. y Honey, P. (1997) *Estilos de aprendizaje.* Bilbao: Mensajero.

2. Calzadilla, M.E. (2002) *Aprendizaje Colaborativo y tecnologías de la Información y de la Comunicación*. Universidad Pedagógica Experimental Libertador. Venezuela. OEI-Revista Iberoamericana de Educación (ISSN: 1681-5653).

3. Coll, C. (1984). "Estructura grupal interacción entre alumnos y aprendizaje escolar." En *Infancia y aprendizaje*. Vol 27-28, pp. 119-138.

4. Coll, C.; Palacios, J.; Marchesi, A. (1995). *Desarrollo psicológico y educación, II*. Madrid: Alianza.

5. Dede, Chris (2000). *Aprendiendo con tecnología*. Ed. Paidos. Bs. As.

5. Fabra, M.L. (1994) *Técnicas de grupo para la cooperación.* Barcelona: Ceac.

6. Gallego, D.J.; Alonso, C.M. y Cantón, I. (1996) *Integración curricular de los recursos Tecnológicos.* Barcelona: Oikos-Tau.

7. Henríquez, P.; RALLO, R.; Gisbert; M. (2002) .*El trabajo colaborativo en la Red: Una herramienta para la innovación en la enseñanza universitaria*. http://users.servicios.retecal.es/sblanco2/html/en_la_universidad.HTM

8. Hernández y Ventura (1992) *La organización del curriculum por proyectos de trabajo.* Barcelona: Graó.

9. Jorba, J. y Casellas, E. (1997*): La regulación y la autorregulación de los aprendizajes*. Editorial Síntesis, Universidad Autónoma de Barcelona, Barcelona.

10. Ovejero, A. (1990) *El aprendizaje cooperativo. Una alternativa eficaz a la enseñanza tradicional.* Barcelona: PPU.

11. Pérez Serrano, G. (1994) "La Educación en el horizonte del Siglo XXI" *Bordón*, 46, 4, 423-440.

12. Prensky, Marc. (2001). *Nativos en inmigrantes digitales"* - NCB University Press, Vol. 9 No. 5,

13. Toledo, P. (1994) *Perspectivas teóricas acerca de los efectos del aprendizaje cooperativo en el rendimiento de los alumnos*. Bordón, 46, 4, 455-462.

14. Tiffin, J. Y Rajasingham, L. (1997): *En busca de la clase virtual. La educación en la sociedad de la información.* Temas de educación. Paidós. Barcelona

2007 Argentinian Congress on Computer Science
(Congreso Argentino en Ciencias de la Computación -CACIC 2007)

Entorno Multimedia: En el Desarrollado de Software Educativos para Aprender y Enseñar Lengua de Señas Argentina

Lic. Dos Santos Romina E.
Licenciada en Sistemas – Programadora de Aplicaciones
FACENA – UNNE - Corrientes, Argentina
rominads@gmail.com

Mgter. Petris Raquel H.
Magíster en Informática y Computación – Licenciada en Sistemas
FACENA – UNNE - Corrientes, Argentina
rpetris@exa.unne.edu.ar

Resumen

Este trabajo de investigación pretende brindar aportes que permitan mejorar el proceso de enseñanza aprendizaje bajo entornos informáticos, en alumnos con ciertas discapacidades auditivas, promoviendo el protagonismo del sujeto, mejorando el conocimiento, facilitando y optimizando el trabajo que supone la tarea de formación respetando los tiempos individuales del aprendizaje.
Para lograr los objetivos propuesto en el diseño y desarrollo de este Software Educativo se recurrió a la elaboración de filmaciones de ejercitaciones de docentes dedicados a las discapacidades auditivas, en la edición de diferentes tipos de videos, construcción de animaciones y gráficos.
Ante la poca disponibilidad de recursos pedagógicos informáticos que pueden ser aplicados en las clases con niños especiales se pretende proporcionar mecanismos que permitan integrar la interactividad que proporcionan las herramientas multimedia con el proceso de enseñanza-aprendizaje, y recrear materiales didáctico con herramientas educativas, aplicando los conceptos modernos de la informática.
De este modo se busca acercar a los estudiantes a esta herramienta moderna y confirmar como la informática ayuda a la construcción de nuevos conocimientos con aplicaciones de fácil manejo, didácticas, amigables al usuario y adaptable a cada situación.

The present research work attempts to give contributions which allow to enhance the teaching-learning process under informatics enviroments, in students who have auditory incapacities, promoting the individual leading role , improving the knowledge, making easier and better the work the formation task supposed to be, respecting the individual learning time.
Films elaborations, different kind of videos, animation and graphics were applied in order to achieve the proposed aims. This Educational Software design is related to teacher's exercises who are devoted to auditory incapacities.
For the lack of pedagogical informatics resources to be applied to special children in class, this Educational Software design attempts to promote strategies which allow to compose the interactivity that multimedia educational tools give with the teaching-llearning process, and thus recreate didatics materials with educational tools, applying modern informatics concepts. in this way, it is searched to approaching students to this modern tool and to stating how informatics helps to the construction of new knowledge with easy operation, didactics which are friendly to the users and adaptable to each particualr situation.

Palabras Claves
Multimedia en la enseñanza-aprendizaje - Entornos Multimedia – Aprendizaje e Informática – Software Educativo – Educación Especial -

Introducción
Quienes no conocen la Lengua de Señas, piensan que las señas que realizan los hipoacúsicos para comunicarse son un conjunto de signos y movimientos pocos claros.
El Lenguaje de Señas Americano (LSA), como cualquier otro idioma, está en permanente desarrollo y evolución, por lo cual se lo puede considerar una lengua viva.
Por esta razón es que se piensa en una manera fácil, práctica, motivadora y atractiva de fusionar la educación con la tecnología y se da inicio al desarrollo del software educativo multimedia que facilite el proceso de enseñanza no solo para los docentes con sus alumnos, sino también para todas aquellas personas que estén interesadas en el estudio de esta lengua.
A través de diferentes indagaciones se logró alcanzar el objetivo general que se planteo para este proyecto; lograr de manera sencilla y gráfica todas las cuestiones que hacen a la LSA; por medio de ejemplos prácticos en una aplicación multimedia, utilizando para ello interfaz, gráficos, menús, animaciones, videos, etc.
A través de este proyecto se intenta dar una visión genérica, pero a la vez detallada, de todo lo relacionado al LSA. Si bien es cierto que para poder utilizar el lenguaje es necesario profundizar en la gramática, la configuración de la sintaxis, los tipos de movimientos, entre otras consideraciones; podemos decir que esto se ha logrado gracias a una visión detallada y se ha analizado cada uno de los pasos que hay que dar para iniciarse en el idioma.
Para el diseño del software, se tuvo presente la naturaleza de las personas que utilizarían el mismo, por lo que se buscó presentar una interfaz amigable y accesible, altamente interactiva e interesante de usar; y con una distribución uniforme, lo que permite al usuario navegar por el software sin desviarse de la información, de tal forma que la aplicación resulta atractiva y práctica de recorrer facilitando de esta forma la elección de los contenidos a estudiar como también la reformulación de los conceptos para profundizar.
Es oportuno aclarar que la Lengua de Señas no es un idioma universal, presenta características y particularidades de la zona donde es utilizado. Si bien en algunos lugares difieren en grandes aspectos en otros existen señas que se consideran aptas para diversas zonas. [1]
La información desarrollada en este proyecto educativo digital se desglosa en dos etapas:

Los Conceptos que hacen al Idioma, como ser:
1- ¿Qué es el Lenguaje?
2- Definición de la Lengua de Señas.
3- Origen de la Lengua de Señas.
4- Historia de la Lengua de Señas Argentina.
5- Comunidad Sorda.
6- Diferencia Entre Lengua de Señas y Otros Sistemas de Señas.
7- Consideraciones Previas Sobre la Lengua de Signos.
8- El Origen del Bilingüismo para los Sordos.
9- La Propuesta Oralista en Relación con los Planteos Lingüísticos.
10- Relación entre La Sociedad y La Educación.
11- Educación Bilingüe - Bicultural.
12- ¿Qué es La Educación Bilingüe Para Los Sordos?

La Estructura Gramatical del Idioma, como ser:
1- La Estructura de la Lengua de Señas.

2- Como Leerle a los Sordos.
3- Estructuras Diferentes para las Lenguas de Señas y las Orales.
4- La Lengua de Señas y Las Lenguas Naturales.
5- Canales de Expresión y Comprensión del Lenguaje de Signos.
6- El Número y El Género en la Lengua de Señas Argentina.
7- La Función Lingüística de Los Rasgos - No Manuales.
8- Ordenes Marcado y No Marcado en LSA.

Desarrollo del Proceso

En cada una de las etapas metodológicas se han considerado los aspectos de naturaleza pedagógico-didácticas. Esto da soporte a un desarrollo tecnológico interdisciplinario, que tiene como pilares a la ciencia informática y a las teorías del aprendizaje.
Al analizar el ámbito que comprende a la Lengua de Señas, se encontró la carencia de información útil que permitía al usuario iniciarse en dicho idioma en forma personalizada.
Otro de los problemas que se presentó, fue la complejidad para entender y estudiar la estructura del idioma. Por ello se decidió incorporar a los contenidos teóricos del software, una ayuda que facilite al usuario comprender los conceptos básicos de la estructura. [7]
Además, se debe considerar que la actualización de la enseñanza es un proceso continuo y asentir que la progresiva introducción de los medios informáticos en las actividades educativas provoca cambios, tanto en la forma de plantear los problemas como en el modo de resolverlos.
El hecho de actualizar los materiales educativos, despierta un interés en los docentes y alumnos por emplear y formarse a través de este nuevo medio de enseñanza, lo que ayuda a innovar y motivar el aprendizaje. [10]
El software que denominamos "*En Otras Palabras*" se encuentra en equilibrio con las tendencias pedagógicas actuales, para que su uso en los entornos educativos provoque un cambio metodológico en este sentido.

Evaluación y Selección de las Herramientas de Desarrollo

Para abordar lo relacionado al diseño del software se han considerado herramientas que permitieran enlazar las páginas hipermedias, y que a su vez tengan una gran compatibilidad con otros productos utilizados, para que en un futuro se pueda ampliar los conceptos de este software. [7]
Se eligió Macromedia Flash para la creación en forma general de las páginas hipermedias. Considerando que este programa permite combinar texto, botones de navegación, gráficos de mapa de bits, gráficos vectoriales, fotos, animaciones, videos, etc. [14] [16]
En la etapa de diseño se desarrollaron los gráficos que se utilizaron para el desarrollo del software para lo cual fue necesario emplear herramienta de dibujo vectorial y de edición de imágenes de mapa de bits. Para esto se recurrió a los programas CorelDRAW que es una aplicación de diseño vectorial, que incluye herramientas para el tratamiento de mapas de bits y además goza de una alta compatibilidad con varios formatos y Corel Photoshop que permite la aplicación informática de edición y retoque de imágenes bitmap. Ambos estándares mundiales que se usa ampliamente en las disciplinas del campo del diseño y fotografía o en otas actividad que requiera el tratamiento de imágenes digitales. [11] [15]
Como segundo paso se considero el diseño de las diferentes pantallas que formarían al software, como así también los contenidos y el método de navegación entre las mismas.
Un menú principal, que contendría todas las categorías a ser comprendidas dentro del software. Por cada uno de los menús considerados anteriormente se diseño un submenú que presentan diferentes animaciones, con la intención de causar un mayor interés en el usuario. A partir de los mismos se podrá acceder a la información. [16]

Cada categoría tendrá los conceptos básicos de la estructura de la Lengua de Señas, como así también una ayuda del manejo del software, algunas noticias de interés general sobre el LSA y una serie de cuentos para ser narrados en LSA a los alumnos. [10]
Los videos utilizados para representar las señas, fueron editados por medio del software Studio de Pínchale, grabados en forma analógica, luego digitalizados y trasladados a CD-ROM para ser exportados al programa Premiere 5.0 de Adobe, con el cual se desarrollo la edición final de los mismos.

Diseño de la Interfaz y la Navegación

Para desarrollar esta aplicación multimedia se consideró en primer lugar *"qué es lo que se quería hacer"*. Lo que permitió fijar las etapas y decidir en que modo se va a expresar la información de cada una de estas partes. Por último se plasmó mediante un diagrama de estructura la aplicación, los enlaces, las nodos principales y como iba a discurrir la navegación y los accesos entre ellos. Ver Figura 1.

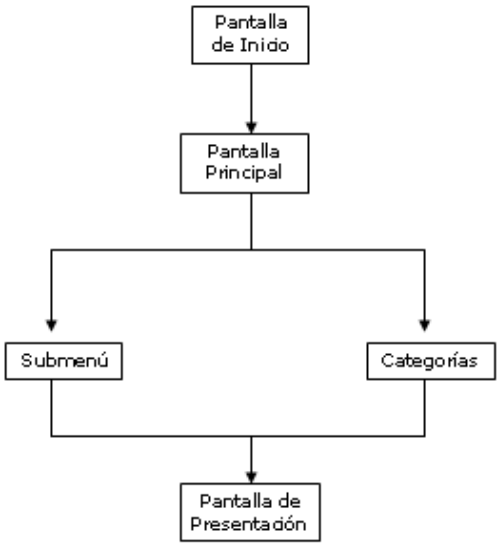


Figura 1

Se logro una aplicación multimedia amigable, intuitiva y flexible al manejo. Como también una adecuada navegación y acceso a la información siguiendo metodologías actualizadas para el diseño y desarrollo multimedia. En nuestro caso hemos optado por una combinación del Acceso Estrella e Indexado. [7]
Para el diseño del esquema de distribución de los objetos en las pantallas que se incluirían en el software, se tuvieron en cuenta los siguientes pasos:
El escenario: Es el lugar físico donde va a transcurrir la aplicación. Puede variar su aspectos según los objetos que se están presentando pero siempre manteniendo la estructura básica. Ver figura 2.

Figura 2

La interfaz: Es la parte más importante de una aplicación, por su impacto con el usuario, por lo que debe resultar intuitiva, fácil de utilizar y agradable a la vista. Ver figura 3.
La interfaz se diseñó siguiendo pautas estándares que se detallan a continuación:
1. La existencia de zonas fijas, para que el usuario las identifique fácilmente.
2. La agrupación de botones conforme a acciones comunes, los cuales aparecen en los mismos puntos de la pantalla y mantienen el mismo comportamiento ante acciones similares.
3. La identificación de pantallas según rango o categoría.
4. La inclusión de botones o zonas que permitan avanzar o retroceder hasta un punto especifico.

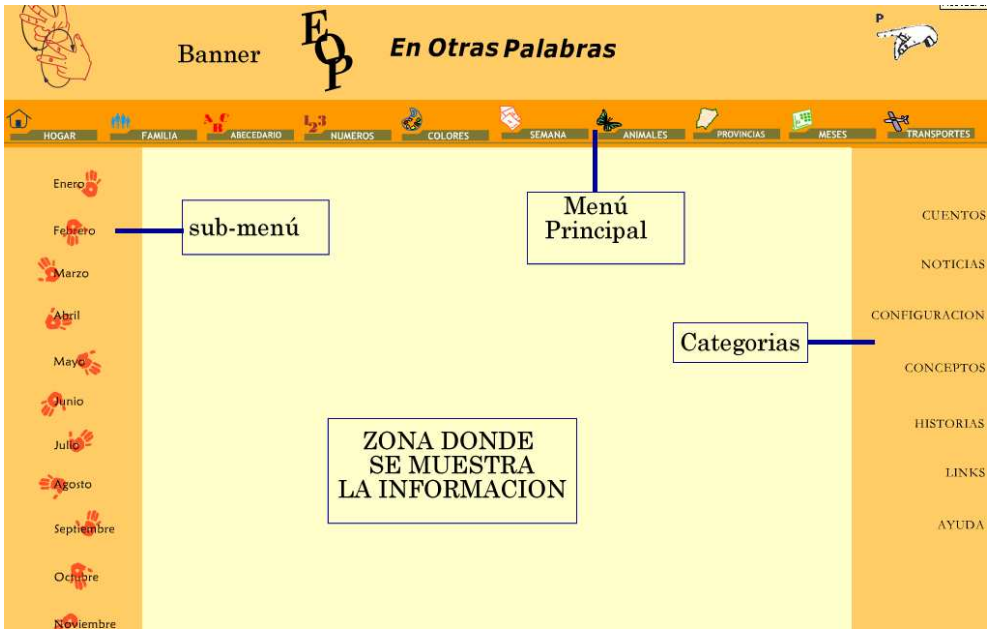


Figura 3

Los Enlaces: cada pantalla contiene enlaces con otras pantallas o zonas de aplicación. Los enlaces particulares deben coincidir con los enlaces genéricos que previamente se definieron en el flujo general de la aplicación. Ver figura 4.

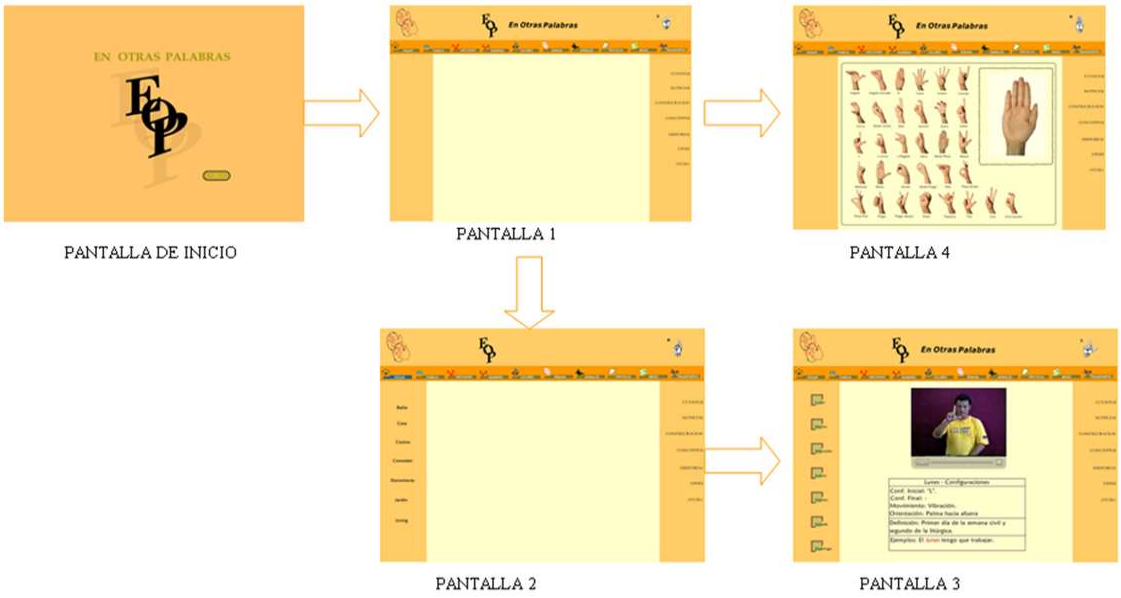


Figura 4

Localización de Zonas Acceso Rápido: La zona acceso rápido de una pantalla o de una interfaz son los puntos en los que va a ocurrir una determinada acción cuando el usuario accione un evento.
Para identificar una zona de acceso rápido, el usuario debe notar un cambio en lo que en ese momento esta viendo. Ver figura 5.
Una vez individualizadas las zonas de acceso rápido, es necesario definir su comportamiento y la acción que van a realizar cuando se les aplique una serie de eventos.

Figura 5

Los Personajes: Personajes u objetos son los elementos que participan en el escenario de una manera activa o pasiva. Se pueden definir en el guión o posteriormente, porque sobre ellos se definirá una serie de acciones posibles, que se llevarán a cabo cuando se invoque a un evento.
Los personajes se diseñan como un elemento alternativo de la interfaz, es decir, no van vinculados con ella, ya que a la hora de programar la aplicación, habrá que animarlos y moverlos, por lo que no pueden ir como parte del fondo.
La Animación: Son secuencias de fotogramas encadenados -en dos o tres dimensiones- que dan sensación de movimiento. Pueden aparecer en una aplicación de distintas formas. Animando a un personaje, presentando una secuencia que sustituya o actúe en lugar de un video, para producir determinados efectos, para animar y enriquecer un determinado evento, siendo siempre un recurso a tener en cuenta. [16]
Se trabajo con gráficos planos, que animados generan los movimientos tradicionales formados por una secuencia continua y en los que cada fotograma tiene una pequeña variación con respecto al anterior, y al reproducirse toda la secuencia de forma consecutiva da una sensación de animación o movimiento.
El Video: Es un aspecto importante en las aplicaciones multimedia, debido a su aporte de realidad.
Los archivos de video que se utilizaron fueron capturados con una cámara de video de tipo analógica y posteriormente fueron digitalizados. Para incluir los archivos dentro de las escenas se utilizó el programa de edición de video Adobe Premiere, el que permitió editar cada una de las secuencias, juntarlas, copiarlas, cortarlas, mezclarlas y renderizarlas.

Validaciones y Difusión del Entorno
En esta fase se comprobó el correcto funcionamiento del software y su entorno multimedia. Esta etapa se elaboró en base a un prototipo del software a desarrollar, el cual consistió en un modelo funcional del producto final.
La depuración del prototipo no solo estuvo destinada a corregir los errores del mismo, sino que sirvió para mejorar el funcionamiento y el rendimiento de algunos aspectos de la aplicación. En esta etapa se contó con la colaboración de personal idóneo tanto en el aspecto pedagógico como curricular. [7]
Los tópicos que se comprobaron y analizaron son:
1. La legibilidad de las imágenes empleadas, así como su correcta ubicación y proporción en tamaño y colores.
2. La legibilidad y corrección de todos los textos.
3. El flujo correcto de la aplicación. Comprobando la secuencia de todos los enlaces.
4. La navegabilidad del software comprobando:
    - La no existencia de zonas muertas en la aplicación, es decir puntos desde donde no se puede acceder a ningún otro sitio.
    - La inexistencia de zonas inaccesibles, es decir a las que no se puede acceder desde ningún punto.
    - La posibilidad de volver hacia atrás o salir desde cualquier punto.
5. La reproducción correcta del video, sin saltos y sin errores.
6. Un adecuado tiempo de ejecución, ni demasiado lento, ni demasiado rápido.
7. Un tamaño de la aplicación apropiado, así como la compresión de cada uno de los archivos.
8. Que la aplicación sea integra y no produzca errores del sistema ni mensajes extraños.
9. Comprobar todos estos pasos en distintos equipos, con configuraciones diferentes.

La aprobación del prototipo permitió continuar con el desarrollo del proyecto hasta finalizarlo, para luego someterlo a otra evaluación, que consistió en analizar aspectos relacionados con la veracidad de la información presentada en los contenidos teóricos su ortografía y sintaxis, las correspondencias entre elementos multimedia y los contenidos teóricos en cada página y

principalmente la comprobación de la correcta navegación. Básicamente, se evaluó, que el software educativo “En Otras Palabras” esté funcionando adecuadamente.
En la última etapa se realizó una prueba piloto del producto final. La cual se basó en presentar el software terminado a un grupo de docentes de alumnos hipoacúsicos para poder evaluar dos cuestiones:
1. Los aspectos pedagógicos, es decir, aquellos que se refieren al fin con el que el software será utilizado.
2. Los aspectos funcionales, en los cuales se consideró por ejemplo los beneficios que obtiene el docente empleando este material didáctico y cómo el mismo facilita el aprendizaje en los alumnos.
Esta etapa de evaluación fue considerada importante, porque permitió obtener sugerencias que optimizaron el proyecto.

Conclusiones

Un software con estas características crea expectativas en el alumno, por la posibilidad de incluir a personas especiales en la utilización y aplicación de herramientas informáticas para su educación y al mismo tiempo entrenarlo para futuras actividades laborales. El resultado de trabajar con este software, a su vez, puede ser entusiasta y servir para que el usuario inicie un nuevo camino no solo en el estudio del lenguaje, sino también por los entornos multimedia.
Se comprobó que el resultado de la utilización del libro y del software, como material de estudio profundizó el idioma en aquellas personas que tienen un conocimiento del tema y que generó expectativas en quienes aún no comprendían la importancia del mismo.
Actualmente el grupo se encuentra evaluando la posibilidad de ampliar los contenidos curriculares y de profundizar otros, a los efectos de satisfacer las expectativas tanto de los docentes como de los alumnos.

El software Educativo “En Otras Palabras” se encuentra registrado como propiedad intelectual ante la CESSI con el expediente Nº 569011.

Referencias

[1] Antonella Buge. “La Educación Del Sordo”. www.sitiodesordos.com.ar/.
[2] Viviana Burad. “Esta Difícil Tarea de Interpretar para las Personas Sordas.” www.sitiodesordos.com.ar
[3] Skiar. “La Educación de los Sordos”. Ediunc, Mendoza, Argentina, 1997.
[4] Mónica Curiel. “El Habla Visual. Lingüística de Las Lenguas de señas.” www.lsargentina.com.ar - 1993.
[5] Ernesto Davis. “La Historia de La Lengua de Señas”. www.sitiodesordos.com.ar.
[6] María Ignacia Massone. “LSA Primer Diccionario Bilingüe”. Sopena, Bs. As., Argentina, 1993.
[7] Nacho B. Martín. “Guía Visual de Multimedia”.
[8] María Ignacia Massone. “El Habla Visual. Lingüística de Las Lenguas de Señas”. www.lsargentina.com.ar - 1993.
[9] Carlos Skliar. “Una Mirada Sobre los Nuevos Movimientos Pedagógicos en la Educación de Los Sordos”. www.sitiodesordos.com.ar
[11] Corel DRAW Graphics Suite 12. www.coreldraw.malavida.com/descarga.
[12] CorelDraw 12. www.aulaclic.com
[13] Fireworks 8. www.adobe.com/es/products/.reworks/
[14] Flash 8. www.aulaclic.com
[15] PhotoShop CS2. www.aulaclic.com
[16] Chris Bedford. Utilización de Flash 8. Macromedia, Inc., San Francisco, EE.UU, 2005.

"Experiencia de laboratorio en Robótica: De las Tortugas Simuladas a los Robots y Juguetes Autónomos una Oportunidad Para la Enseñanza de Programación"

J. Ierache [(1)] M. Bruno [(1)], N. Mazza [(1)]
[(1)]Instituto de Sistemas Inteligentes y Enseñanza experimental de la Robótica (ISIER)
Facultad de Informática Ciencia de la Comunicación y Técnicas Especiales
Universidad de Morón Cabildo 134, (B1708JPD) Morón, Buenos Aires, Argentina
54-11-56272000 interno 189/746
[(1)] {jierache, mabruno, nmazza} @unimoron.edu.ar

Abstract

The experiences are framed in the context of the Technologies in Education, in the matter of robotics, those appear specifically that contribute to the conformation of a laboratory in robotics, integrated basically by robots classic, biped and quadruped which they are in the category of independent toys, this last characteristic facilitated an ample access to different universities, institutes and individuals of different parts from the world motivated by his low cost and the documentation available, generating the development of sensors, specific environments of programming in C, JAVA, among other languages, like thus also specific developments for necessary firmware. Against this background, the education of programming applied through physical agents robots allows the interaction between agents and of these with the performance atmosphere. The tools presented with the application of robots physical (Lego RCX, NXT, Robosapien, Robopet) compared with the simulated turtles (logo), present a strategic opportunity for the formation of human resources.

Key words: Robotics, Technologies in Education, Programming, Intelligent Systems, Agents and Multiagents

Resumen

Las experiencias se enmarcan en el contexto de las Tecnologías en Educación, en materia de robótica, se presentan específicamente aquellas que contribuyen a la conformación de un laboratorio en robótica, integrado básicamente por robots clásicos, bípedos y cuadrúpedos que se encuentran en la categoría de juguetes autónomos, esta última característica facilitó un amplio acceso a distintas universidades, institutos y particulares de distintas partes del mundo motivados por su bajo costo y la documentación disponible, generando el desarrollo de sensores, ambientes específicos de programación en C, JAVA, entre otros lenguajes, como así también desarrollos específicos para el necesario firmware. En este contexto, la enseñanza de programación aplicada a través de agentes robots físicos permite la interacción entre agentes y de estos con el ambiente de actuación. Las herramientas presentadas con la aplicación de robots físicos (Lego RCX, NXT, Robosapien, Robopet) comparadas con las tortugas simuladas (logo), presenta una oportunidad estratégica para la formación de recursos humanos.

Palabras claves: Robótica, Tecnologías en Educación, Programación, Sistemas Inteligentes, Agentes y Multiagentes.

1. INTRODUCCION

La emoción de ver como una tortuga deambula en nuestro monitor esquivando obstáculos virtuales hasta llegar a su meta en la esquina de un monitor [16] no tiene el mismo impacto emocional que

observar como un agente robot implementado sobre la base de juguetes autónomos puede esquivar obstáculos hasta llegar a su meta en el rincón de una habitación e interaccionar con nosotros a través de nuestro celular. Existen proyectos relacionados entre los que caben destacar el que se lleva a cabo en la Universidad Estatal de Campinas (UNICAMP); Brasil, donde se utiliza como base al lenguaje Logo con fines educacionales [27]. En particular, trabajan con SuperLogo, una versión del lenguaje LOGO desarrollada por el NIED, que permite el control del ladrillo programable LEGO RCX, que reproduce los movimientos de la tortuga en pantalla. El control del RCX se puede hacer por medio de una PC, en forma local o remota, utilizando la Internet. En la Universidad de Medford/Somerville en Massachussets, utilizan Lego RCX, Robolab y LabView para la enseñaza con distintos niveles de complejidad [24]. Por ejemplo en las clases de Introducción a la Ingeniería Mecánica. También los utilizan en cursos de robótica para estudiantes de primer año y en cursos avanzados de robótica con desarrollo de algoritmos de alto nivel e inteligencia distribuida.
Muchos proyectos involucran un control centralizado clásico, la computadora le dice al motor uno que se encienda, gire en sentido de las agujas del reloj, a media potencia bajo una secuencia planificada de acciones, pero también se puede aplicar el mismo agente robot para explorar sistemas descentralizados y de conductas autoorganizadas [1], [2], [28], [29]. Por ejemplo si consideramos un agente que deambula por su hábitat el que cuenta con áreas iluminadas y áreas oscuras, nuestro agente cuenta con dos reglas, una que le indica al agente moverse hacia delante cuando detecta áreas iluminadas y moverse hacia atrás cuando detecta áreas oscuras, el agente deambula hasta que llega a una sombra, entonces retrocede hasta que sale de ella y luego hacia delante otra vez, sigue así oscilando en el borde de la sombra; en este caso podemos considerar a nuestro agente robot como un criatura que detecta bordes, en si esta capacidad no esta explícitamente declarada en sus dos reglas, en realidad es una conducta grupal que surge de la interacción de las dos reglas, algo similar a como la conducta de una bandada surge de la interacción entre los pájaros [22].
Los estudiantes tienden a considerar a sus criaturas según diferentes niveles en distintos momentos, a veces ven a sus criaturas a un nivel mecanicista, al analizar como una pieza mueve a otra, otras veces pasa a nivel de información y exploran como pasa la información entre la computadora y los motores y sensores. En otras ocasiones los estudiantes consideran a las criaturas en un nivel psicológico, atribuyendo a la criatura intencionalidad o personalidad. Una criatura quiere ir a la luz, a otra criatura le gusta la oscuridad, otra le teme a los ruidos, los estudiantes pasan rápidamente entre estos niveles y aprenden en función de la situación de contexto que nivel es el mejor, piensan en sistemas en termino de niveles [22]. La idea de aprender mediante el diseño es un aspecto de lo que Seymour Papert [19] denominó como enfoque construccionista del aprendizaje y la educación. Los seres humanos construyen su conocimiento con particular eficacia cuando participan en la construcción de productos que son afectivos.

2. CARACTERÍSTICAS DE LOS AGENTES ROBOTS Y ENTORNOS DE PROGRAMACIÓN

El objetivo de nuestro trabajo es brindar una orientación general de los Kits de Robótica económicos actuales de Lego RCX y el reciente NXT, sus herramientas de programación en el ambiente de LabView, Robot C, NQC, entre otras. En materia de Juguetes autónomos se detallaran las características más relevantes de “Robosapien”. (robot bípedo) y de “Robopet, Robotail, Roboraptor”(robots cuadrúpedos), se considera además las interfases de comunicación y herramientas de programación, en particular GoRobo.

2.1- NXT y RCX, los robots de Legomidstorms

NXT, la nueva generación de robots de Legomindstorms [14] (Figura 1) ofrece características mejoradas con respecto a su anterior versión el RCX. En una comparación de las principales características de ambos kits se pueden mencionar:

Tabla Características de RCX Vs NXT

	LegoMidstorms RCX	Legomidstorms NXT
Ladrillo inteligente	Ladrillo RCX, con 5 slots para guardar programas del usuario.	Ladrillo NTX, que tiene una mayor capacidad de cómputo que el ladrillo RCX. Incluye funciones para testear los sensores (función Try me). Permite la personalización de los sonidos que el NXT puede reproducir, etc.
Puertos para Motores/ Sensores	Tres puertos para motores	Tres puertos para motores.
	Tres puertos para sensores	Cuatro puertos para sensores.
Sensores/ Motores	1 Sensor de luz, que permite distinguir diferentes niveles de luminosidad y oscuridad. 2 Sensores de tacto, que pueden detectar tres estados: o Presionado. o Libre. o Rebote.	1 Sensor de Luz 1 Sensor de Sonido, que tiene dos modos de configuración: o Para detectar sonidos perceptibles por el oído humano. o Para detectar además los sonidos no perceptibles por el oído humano (como ultrasonidos). 1 Sensor de Ultrasonido que actúa como un radar permitiendo la detección de objetos. Se lo puede configurar para detectar objetos lejanos o cercanos. Detecta objetos entre 0 y 255 cm con una precisión de +- 3 cm. 2 Sensores de tacto, que pueden detectar tres estados: o Presionado. o Libre. o Rebote.
	3 Motores.	3 Servomotores, mejorados con respecto a la versión RCX, ya que tienen integrados sensores de rotación. De este modo se pueden lograr movimientos mucho más precisos y controlados, además de que los dos motores se puedan sincronizar perfectamente.
Comunicación/ Descarga de programas	La descarga de programas se hace a través de la torre de infrarrojos que incluye el kit. La comunicación con otros RCX es través de infrarrojo también	El NXT posee un puerto USB, para la descarga de programas. Este kit incluye además Comunicación Bluetooth inalámbrica, permitiendo descargar programas de este modo, así como también la interacción con celulares y también con PC y laptops, etc. Permitiendo utilizar la capacidad de cómputo de otros equipos. La comunicación con otros NXT también es vía bluetooth
Otras características	1 parlante.	1 parlante de alta fidelidad, mejorado con respecto al del RCX.
	conexiones cableadas	Las conexiones para sensores y servomotores son RJ11.

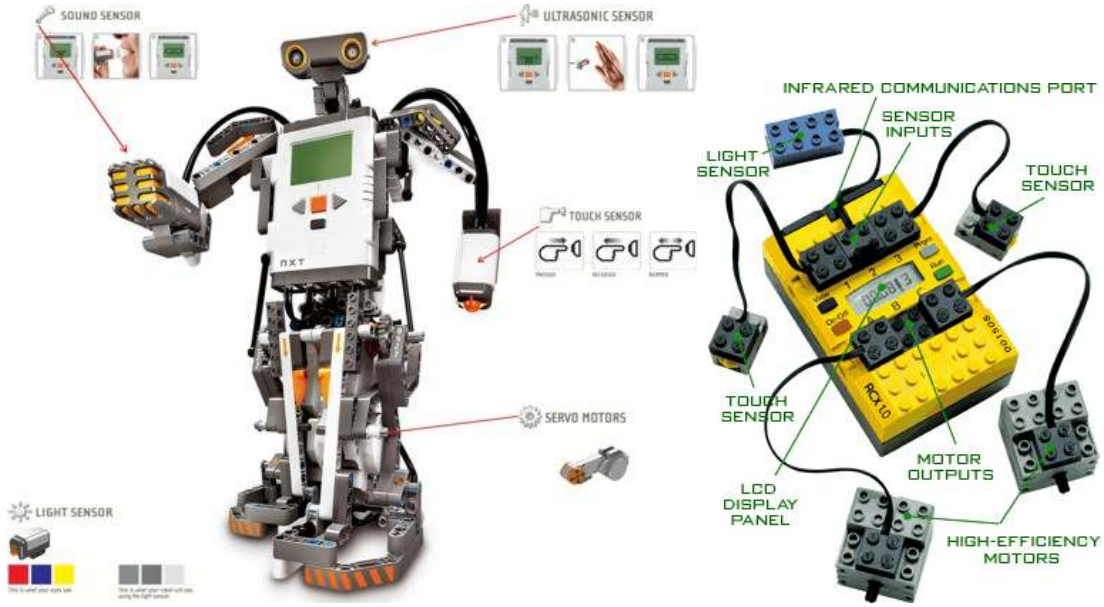


Figura 1: Legomindstorms NXT (izq) y RCX (der)

Estos agentes robots son programables en entornos gráficos nativos, para RCX [14] (figura 2, izq) y LabView [17] para NXT (figura 2, der). En cuanto a LabView cabe destacar que fue desarrollado por National Instruments y utilizado por la NASA, para monitorear y controlar al robot Sojourner Rover, en la misión de exploración de la superficie de Marte [17].

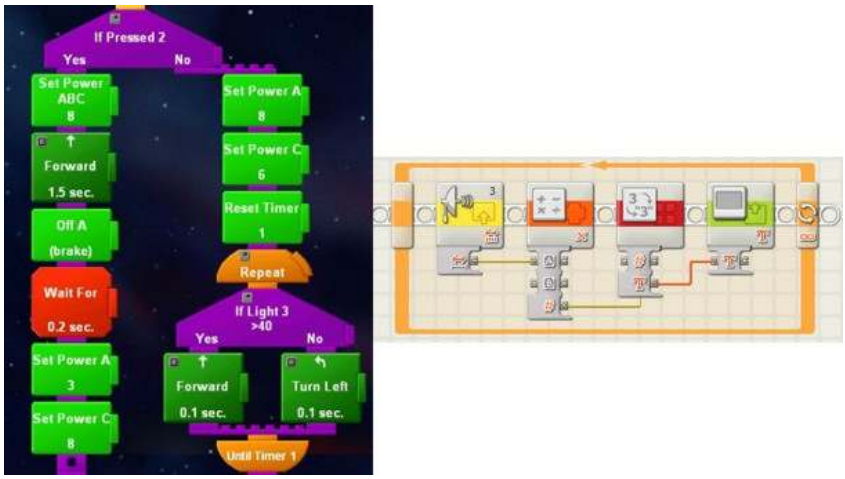


Figura 2: Ambiente de programación de RCX (izq) y LabView (der)

Estos entornos utilizan bloques que se ensamblan para formar un programa completo. Estos bloques incluyen control de motores (avance, retroceso, encendido y apagado), ciclos repetitivos (while, repeat), estructuras de control (if else), adquisición de datos de los sensores, utilización de variables, constantes y timers. Además de estos entonos gráficos existen una serie de programas que permiten su programación en códigos más tradicionales, como por ejemplo en Java. Es el caso del API Lejos para RCX y iCommand para NXT [15]. Uno de los programas más utilizados y que aumenta enormemente las posibilidades de programación es el NQC [1], [2] de Dave Baum, utilizado para programar al RCX en un lenguaje similar al C. Para el NXT, existe un programa llamado RoboC [20], que es mucho más completo que el NQC del RCX, y que además incluye un firmware propio,

que lo hace muy potente. A continuación se enumeran las características más importantes de las principales herramientas de programación.

2.1.1- Lejos Java for Legomindstorms

- Es un nuevo firmware alternativo para el NXT.
- Permite realizar programas en JAVA para controlar robots.
- Funciona en entorno windows y Linux y permite comunicarse con el NXT vía USB.
- En la primera versión todavía no hay soporte para comunicación por Bluetooth, manejo de sensores que se comuniquen por I2C (como el sensor de ultrasonidos) o el soporte de sonido.
- La versión Alpha 0.2 contiene soporte preliminar para bluetooth, soporte para sensor de ultrasonido (I2C) y soporte para el sonido.
- Lejos es el API de JAVA para programar el RCX.
- Su última actualización fue Lejos 3.0
- El nuevo API de JAVA para NXT, se llama iCommand, del cual se acaba de publicar la versión 0.5 Que incluye entre otras cosas Soporte para Webcam y Soporte para brújula electrónica

2.1.2- LabView (entrono grafico de programación)

- Entorno nativo para programar al NXT. Viene junto con el kit de NXT (figura 2, der).
- Utiliza un lenguaje icónico de programación, muy intuitivo.
- Se baja el programa de la PC al NXT por medio del USB o bluetooth.
- Se pueden crear bloques propios
- Comunicación con Bluetooth: el firmware del NXT permite una configuración de tipo Amo-Esclavo para la comunicación con bluetooth.Con bluetooth se pueden comunicar hasta 3 NXT.

2.1.3- RobotC

- Es similar al NQC para los RCX, pero mucho más poderoso.
- Permite programar a los robots en un C reducido
- Incluye un firmware propio.
- Por el momento está disponible la versión beta que es gratuita por 30 días.
- Tiene soporte para la comunicación con bluetooth
- Este es uno de los mejores programas que existen actualmente para desarrollar con LegoMindstorms NXT.

2.2 Juguetes autónomos

Aunque se venden como juguetes, ofrecen unas prestaciones tan avanzadas que hacen de estos un medio excelente para experimentar con la robótica. Así el Robosapien [23], [30] cuenta con siete motores que le proporcionan un número sorprendente de grados de libertad, además de sensores de contacto en los dedos y los pies, y sensor de sonido. Niveles múltiples de interacción ambiental, con humanos y objetos, vista, sonido y sensores de tacto. Movimientos humanoides, que incluyen agacharse, sentarse y pararse, acostarse, levantarse, artes marciales, trucos, etc.

2.2.1- Bípedos

Estos robots, poseen sensores de sonido estéreo, visión infrarroja, y también sensores de tacto para detectar obstáculos y un importante número de grados de libertad. Dentro de esta categoría encontramos:

- Robosapien V1 es una versión con menores prestaciones a nivel de sensores que el V2, no incorpora visión, las capacidades de desplazamiento son similares en sus funcionalidades, si bien la V1 al ser mas pequeña tiene un mejor desplazamiento.
- Robosapien V2, además de las características antes mencionadas incluye también sensores táctiles en los guantes, y en las palmas de las manos, (Figura 3, izq) lo que le permite tomar objetos con ellas. Además posee una cámara que le permite reconocer colores [23].
- Robosapien Multimedia, [23], [30] aumenta aún más las capacidades de RSV2, incluyendo como característica más importante una memoria mini SD, en donde se lo puede programar directamente, a través de un editor de código de tipo gráfico, que es propio de esta versión. Por default tiene cuatro personalidades que el usuario puede modificar. Además puede grabar videos, sonidos en formato mp3, tomar fotos, y luego reproducirlos en su display LCD

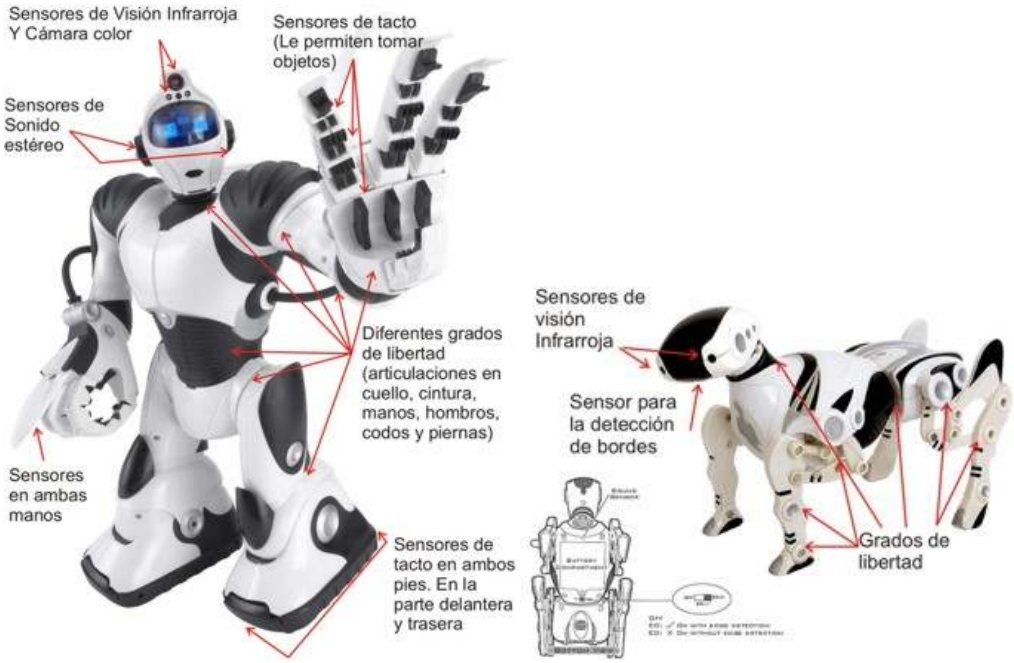


Figura 3: Robosapien V2 (izq), Robopet (der)

2.2.2- Cuadrúpedos

Estos robots, también están equipados con visión infrarroja, sensores de sonido estéreo y motores Entre estos se destacan:

- Robopet, (Figura 3, der) además de lo dicho anteriormente es capaz de interactuar con Robosapien, y también puede detectar los bordes de por ejemplo una mesa [23].
- Robotail, (Figura 4, en verde) posee un sensor de tacto en su lomo, que cuando es presionado, hace que tenga comportamientos diferentes. Además cuando está "hambriento", se vuelve muy "agresivo", y solo se calma "al encontrar comida" [23].
- Roboraptor, (Figura 4) es el roboreptil [23] que también es capaz de interactuar con Robosapien.

Figura 4: Roboraptor y Robotail

2.2.3- Entornos de Programación " GoRobo".

Es un entorno de programación [11], que permite controlar a todos los robots mencionados de la familia WowWee [31]. (Roboraptor, Robopet, Roboreptile, RSV2 y RS Multimedia).

El lenguaje de programación que utiliza se denomina GRIDscript (Go-Robo ID script) [12]. Utiliza una sintaxis de programación simple y consistente basado en las prácticas modernas de productos de programación comerciales (Visual Basic, C++, etc). GRIDscript [12] utiliza sintaxis básica de programación (While/EndWhile, For, If/Else/Endif, Repeat/EndRepeat), la creación de procedimien -tos y el uso de variables.

El alumno puede utilizar este lenguaje para definir procedimientos simples que, posteriormente, pueden combinarse para crear otros más complejos. Además, los robots pueden programarse para interactuar entre sí, ya que el software permite el control de seis de ellos de forma simultánea. Las órdenes se transmiten a través de una torre infrarroja que a modo de intérprete envía los comandos a cada robot, distinguido por su tipo. La versión actual de GoRobo [12], no soporta la torre infrarroja de LegoMindstorms. Solo soporta USB-UIRT [26]. y RedRat3 [21]. A lo largo de este año, sus desarrolladores planean extender el rango de los dispositivos soportados, para incluir a los IR. Dado que por el momento RSV2, no da ningún feedback a GRIDscript, hay una limitación en cuanto a lo que se puede hace con los sensores. Algunas de las cosas que se pueden hacer con GoRobo son: utilizar bloques condicionales y de repetición de instrucciones, utilizar eventos condicionados por timers, se puede introducir ejecución aleatoria de código.Este lenguaje, se ha diseñado para adaptarse a cualquier edad y utilizarse tanto en un contexto educativo como en uno más profesional, donde se genera un ambiente de interacción de los lenguajes clásicos y formales de programación con los comandos nativos de los robots utilizados.

2.2.4- Otras opciones de Programación

Tanto Robosapien como Robopet, reciben los comandos vía IR por control remoto, de este modo hay quienes han realizado un mapeo de dichos comandos a hexadecimal [11], esto hace posible la programación de Robosapien descargando el código con la torre de IR de Lego Mindstorms. El problema es que hay una limitante con la cantidad de instrucciones que el RS puede recibir, que son veinte como máximo. Otra opción más radical: El transplante de cerebro al Robosapien, hay quienes han optado por reemplazar la cabeza del Robosapien V1 por una Palm [23], de este modo, se elimina el problema de la cantidad de instrucciones que se pueden enviar al RS, ampliando enormemente la capacidad de cómputo para Robosapien V1.

3. DESARROLLO AVANZADO PARA AGENTES ROBOTS

La programación de robots presenta interesantes desafíos especialmente en el desarrollo de aplicaciones que actúan en ambientes dinámicos, cooperativos, la elaboración de estrategias para alcanzar sus metas, con la oposición de contrincantes, como así también la capacidad de reaccionar ante situaciones no consideradas en la programación de su control, sin actuación de supervisión

exterior. Su programa de control no debe definir de modo explícito todas las posibles acciones ante todas las posibles situaciones que se pueden presentar en su entorno.
El robot no debe ser totalmente preprogramado [5], [6], [7], [8], [9] debe poseer una arquitectura cognitiva que permita una relación entre sus entradas sensoriales y sus acciones sobre el ambiente, capacidad de generar su mapa autónomo de sensorizaciones - acciones para sobrevivir y alcanzar sus objetivos. El robot debería en su tiempo de vida realizar cada vez mejor sus tareas, aprendiendo por ejemplo de sus errores. El ambiente dinámico, facilita la evolución en función de las distintas sensorizaciones de cada robot, sus metas y la interacción con agentes humanos. Se presenta un escenario favorable para el desarrollo de aplicaciones centradas en contexto donde la participación de actores robots y humanos resulte de interés en un entorno cooperativo
El trabajo de programación del robot además debe desarrollar la interacción con humanos y sistemas de información apoyando la participación de mascotas robots integradas con las tecnologías de información para facilitar la comunicación con humanos en ambientes dinámicos y distribuidos.
La estrategia inicial de comunicaciones para apoyar la interacción entre los agentes autónomos y humanos se basa en el empleo de las facilidades de comunicación inalámbrica Bluetooth entre los agentes robots NXT y dispositivos manuales (celulares, palm, etc). En relación a la familia Robosapien esta se integrará a través de IR en la primera etapa con comunicaciones desde el procesador local de ambiente, el que actuará de gateway entre el mundo de comunicaciones IR y el de Comunicación Bluetooth para el escenario local, este gateway en principio facilitará la interacción de agentes en distintos escenarios distribuidos.
Se busca permitir la interacción de los agentes: NXT, como agente clásico, Robosapien V2 como bípedo y robopet como cuadrúpedo, en un escenario conformado por un Ambiente Dinámico, que en este caso será fútbol de robots, donde los Agentes Robots Autónomos interactúen entre sí a fin de facilitar el desarrollo e investigación en robótica autónoma.
Estos agentes se han seleccionado porque además de tener un costo relativamente bajo con respecto a otros robots, poseen cualidades que permiten explotar sus capacidades al máximo, fomentando el estímulo del aprendizaje en los estudiantes. Los jóvenes se ven atraídos de inmediato por estos robots y las cosas que son capaces de hacer. En su deseo de que se comporten de determinadas maneras, tal vez como hasta el momento pensaban que era posible solo en las películas de ciencia ficción, los hace interesar en temas como la programación, los conceptos de agentes y multiagentes, visión artificial, adquisición de datos, etc. De este modo, se puede aprovechar el interés de los jóvenes en estos robots, para utilizarlo como el motor impulsor, que genera la necesidad de aprender nuevos conceptos que los ayuden a programar sus robots. Por ejemplo, en el caso de los robots de la familia WowWee [23], [30], se tiene un entorno de programación sencillo, que permite introducir al estudiante en los conceptos básicos de la programación, como estructuras de control, bloques condicionales, eventos temporales, etc. Con GoRobo [12], el estudiante puede programar una secuencia de acciones, para el Robosapien V2, para el Robopet o Robotail, y controlarlos en forma simultánea a través de una torre USB-UIRT, en este caso. En este nivel de programación, no se actúa directamente sobre los sensores y motores de los robots, sino que se envían instrucciones que engloban a todas estas acciones de forma transparente para el usuario. Sin embargo, los estudiantes más avanzados, pueden programar al robot NXT de Legomindstoms [14]. En este caso, para su programación, se actúa directamente sobre los sensores y los motores. Además con el NXT, el estudiante puede dar vuelo a su imaginación y modificar el diseño de su robot cuantas veces quiera. De este modo puede construir desde un bípedo, hasta una mano robótica, un robot motorizado, etc.
En cuanto a los entornos de programación [5], estos pueden ser gráficos del tipo de LABView, este ultimo específicamente fue aplicado por la NASA, para la adquisición de datos y control de robots como Sojourner Rover en la misión de exploración de la superficie marciana [17]. o de código

como RobotC [20], iCommand [15], etc. En ambos casos y como se dijo anteriormente, la programación debe actuar directamente sobre los motores, en base a los valores medidos por los sensores, etc. En este nivel, el estudiante adquiere conocimientos de programación de más bajo nivel que en el anterior, y los códigos resultantes con mucho más complejos que en el caso de los robots de la familia WowWee, pero al mismo tiempo, las posibilidades de lo que puede hacer crecen enormemente. Por otra parte, se pretende brindar información global del ambiente a través de la integración de un sistema de visión, que permita la detección y localización de objetos y agentes en el escenario, en este caso el nivel de complejidad se aumenta aún más, posibilitando, que el robot, además de procesar la información de sus propios sensores, tenga información de todo lo que pasa a su alrededor. De este modo el objetivo final del proyecto es el de facilitar a los agentes autónomos (robots y humanos), interactuar entre sí en un escenario bajo un ambiente dinámico, basándose en la información global del ambiente y la propia información del agente obtenida de sus sensores. Es en este nivel, donde se logra la mayor complejidad, y donde se abarca la mayor cantidad de conceptos. A esta altura, el estudiante debe estar familiarizado con los conceptos de agentes y multiagentes, que tienen sus propias metas, objetivos y creencias, que son capaces de interactuar entre sí, e inclusive cooperar para lograr un objetivo común, (en el caso del fútbol, anotar la mayor cantidad de goles, y evitar que el equipo contrario haga goles). También se familiarizará con los conceptos de inteligencia artificial, que posibiliten que los robots, tengan un comportamiento autónomo que varíe en función de la situación, y que no esté preprogramado, sino que se adapte dinámicamente en función de lo que es más conveniente. Y por último, deberá familiarizarse con los conceptos de visión artificial y procesamiento distribuido, que son necesarios para que cada robot, tenga un apoyo de información global, para mejorar la toma de decisiones.

4. NUESTROS AGENTES ROBOTS

Hasta el momento en el Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robotica (ISIER), se cuenta con los siguientes robots: tres RCX, un NXT, un Robosapien V2, un Robosapien V1,un Robopet y in Robotail También los siguientes softwares: GoRobo con licencia Home Edition, RobotC, en su versión beta, iCommand en su versión beta. Hasta el momento se ha conseguido controlar simultáneamente a los tres robots de la familia WowWee (al RSV2, Robopet y Robotail), con el programa GoRobo, utilizando para ello, una torre USB-UIRT de infrarrojos, y también se ha hecho interactuar al RSV2 con el Robopet (Figura 6, izq).

Figura 6: Interacción de RSV2, Robopet y Robotail (izq), NXT siguiendo la línea negra (der)

En cuanto al NXT, se lo programó con LabView, el entorno nativo, con programas simples como por ejemplo el de seguir la línea negra (Figura 6, der). También se lo programó con RobotC, previa bajada del firmware de dicho software. En este caso se simuló el comportamiento de un animal

herbívoro, deambulando por su hábitat, que en este caso fue una alfombra delimitada por una estructura de madera, con papeles verdes distribuidos aleatoriamente, que representaban comida. Así este herbívoro, deambulaba tranquilamente, hasta que encontraba un sector de comida. En este punto se detenía a "comer", además era capaz de detectar los bordes del hábitat, gracias a su sensor de tacto y esquivarlos. En el momento que detectaba el sonido de un predador (con el sensor de sonido), que en este caso fue el Robotail, o que algo se posicionaba por detrás (cosa que detectaba con el sensor de ultrasonido), se "asustaba" dejando de comer y comenzando a escapar a velocidad. En este caso, se pueden observar claramente dos tipos de comportamiento, uno de búsqueda de comida, y otro de escape, que dependen de la interacción del NXT, con su entorno y con robotail (Figura 7).

Figura 7: NXT interactuando con el ambiente y Robotail

5. CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

Se presentaron las características mas significativas de distintos robots y juguetes autónomos de bajo costo, acompañados de sus lenguajes de programación que facilitan el desarrollo de experiencia en el área de la robótica y contribuyen a la enseñanza de programación de alumnos principiantes, como así también facilitan por parte de alumnos avanzados el desarrollo de agentes autónomos, orientando al desarrollo en este caso a una amplia variedad de modelos computacionales descentralizados, como por ejemplo, las redes neuronales, arquitecturas de subsunción y autómatas celulares, entre otros. Finalmente se presenta la base inicial para una arquitectura que facilite la interacción de agentes en ambientes dinámicos, en este contexto las líneas futuras de investigación se orientan al desarrollo de framework de integración de los distintos robots, el desarrollo de capacidades de simulación interoperable entre mundos virtuales y reales a fin de enriquecer el escenario de aprendizaje de los robots, el procesamiento distribuido del ambiente entre agentes autónomos y la integración de agentes robots autónomos en hábitat inteligentes.

Agradecimientos: Este trabajo ha sido soportado en parte por el proyecto A01-007/06 Secretaria de Ciencia y Tecnología de UM y la FICCTE.

6. BIBLIOGRAFÍA

[1]. BAUM, Dave. *NQC Manual.* [en línea]. [ref. 2 de Agosto de 2006]. Disponible en Web: <http://bricxcc.sourceforge.net/nqc/doc/NQC_Manual.pdf>

[2]. BAUM, Dave; HANSEN, John. *NQC Programmer's Guide*. [en línea]. Versión 3.1 r5. [ref. 2 de Agosto de 2006]. Disponible en Web: < http://bricxcc.sourceforge.net/nqc/doc/NQC_Guide.pdf>

[3]. BEARD, R. *"Ball prediction for Robot Soccer*". Department of Electrical & Computer Engineering, Brigham Young University, Provo. 2003

[4]. BILL, N; SCHILIT, Norman Adams; et. al. "Context-Aware Computing Applications, *IEEE Workshop on Mobile Computing Systems and Application*, December 1994.

[5]. BOER, R; KOK, J. "*The Incremental Development of a Synthetic Multi-Agent System:The UvA Trilearn 2001 Robotic Soccer Simulation Team*". Faculty of Science, University of Ámsterdam. 2002

[6]. BORENSTEIN, J; EVERETT, H. R; FENG, L. *Navigating Movile Robots: Systems and Techniques*. 1996

[7]. BRATMAN, Michael E. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, Mass., 1987.

[8]. CASTELO, C; FARSI, H; SCARPETTINI, F. Tesis de Licenciatura "*Fútbol de Robots: Revisión del Estado del Arte y Desarrollo del Equipo UBASot de Simulación*". Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires. 2002

[9]. CLARO, M.; GAZOLLI, A; POTENZA, A; VISCUSO, Germán; IERACHE, Jorge. "*El equipo Morasot*". Facultad de Informática, Universidad de Morón. 2003

[10]. CompuBlog, El blog del aula de robótica Compubot. EducaMadrid [en línea]. [ref. 10 de Abril de 2006]. Disponible en Web:<http://complubot.educa.madrid.org/blog/>

[11] Controlling RoboSapien using LEGO IR-Tower. Trondheim-Bratislava, July 2005 - January 2006. Diponible en Web: <http://www.robotika.sk/maine.php>

[12]. GOROBO, [en línea]. , [ref. 20 de Abril de 2007]. Disponible en Web: <http://www.q4tecnologies.com/>

[13]. GUESTRIN, C; VENKATARAMAN, S; KOLLER, D. *Context Specifffic Multiagent Coordination and Planning with Factored MDPs*. In Proceedings of the Eighteenth National Conference on Artificial Intelligence, Edmonton, Canada, July 2002.

[14]. LEGO.com. MINDSTORMS NXT Home. [en línea]. [ref. 1 de Abril de 2007]. Disponible en Web: <http://mindstorms.lego.com/default.aspx>

[15]. LeJOS, Java for Legomindstorms. SourceForge [en línea]. [ref. 10 de Abril de 2007]. Disponible en Web: <http://lejos.sourceforge.net/>

[16] MOROLLÓN SÁNCHEZ, María; SEGOVIANO, Alonso. *1, 2, 3... Logo (Ideas e Imaginación)*. Centro de Orientación de Sociología y Psicología Aplicada. Madrid, España: Cospa, 1985.

[17]. Nacional Instruments. *LabVIEW*. [en línea]. [ref. 1 de Abril de 2007]. Disponible en Web: <http://www.ni.com/academic/mindstorms/>

[16]. NQC – Not Quite C. [en línea]. [ref. 2 de Agosto de 2006]. Disponible en Web: <http://bricxcc.sourceforge.net/nqc/index.html>

[19]. Papert, S. Situating constructionism, en I. Harel y S. Papert (comps.),Constructionism.Norwood,NJ,Abel Publishing. 1991.

[20]. QUICK START GUIDE, [en línea]. Robotics Academy, Carnegie Mellon University, [ref. 5 de Abril de 2007]. Disponible en Web: < http://www.robotc.net/>

[21]. RedRat3 – USB Universal Remote Control for PC. Disponible en Web: <http://www.redrat.co.uk/RedRat3/index.html>

[22]. Resnick Mitchel Tortugas, Termitas y Atascos de Tráfico, Gedisa, 2001

[23]. ROBOSAPIEN.tk, the first unofficial robosapien hacks and mods site. [en línea]. [ref. 2 de Agosto de 2006]. Disponible en Web: <http://home.planet.nl/~pruim006/main.htm>

[24]. ROGER, Chris. *LEGOS, ROBOLAB, and LabVIEW: Designing, Programming, and Collecting Data*. University of Wisconsin-Madison.The Institute on Learning Technology. National Institute of Science Education. EEUU. 2001. Disponible en Web: <http://www.wcer.wisc.edu/archive/cl1/ilt/extra/download/solution/rogerscw97.doc>

[25]. RUSSELL, Stuart; NORVIG, Peter. *Inteligencia artificial. Un enfoque moderno*. GUTIÉRREZ, Raúl Bautista (trad.); RANGEL, Raymundo Hugo (rev. Tec.). 1a ed. México: Prentice Hall Hispanoamericana, 1996. 979p. ISBN: 968-880-682-X

[26]. USB-UIRT Home Page. Disponible en Web: <http://www.usbuirt.com/>

[27]. VIEGAS D'ABREU, João Vilhete; CHELLA, Marco Túlio. Superlogo-RCX Ambiente para Robótica Educacional. Disponible en Web: <http://www.nied.unicamp.br/~siros/doc/pedagogia_2005_cuba.pdf>

[28]. WOOLDRIGE, Michael; JENNINGS, Nick. R. *Agent Theories, Architectures and Languages: a Survey in Eds. Intelligence Agents*. Berlin: Springer-Verlag, 1995. Vol 1, Nro 22

[29]. WOOLDRIGE, Michael. *An introduction to Multiagent Systems*. John Wiley & Sons. 2002.

[30]. WowWee Robotics. RS Multimedia. [en línea]. [ref. 10 de Abril de 2006]. Disponible en Web:< http://www.rsmediaonline.com/>

[31]. WowWee Robotics.. [en línea]. [ref. 10 de Abril de 2006]. Disponible en Web: <http://www.woowee.com>

Ferramenta para Classificação dos tipos de Transtornos por Déficit de Atenção/Hiperatividade

Robert Mady Nunes[1], Camila Ramos Alves[1], Thiago Moreira Alves[1]

Fernando Luiz de Oliveira[1], Cristiane de Quadros Mansanera[2]

[1]Curso de Sistemas de Informação – Centro Universitário Luterano de Palmas (CEULP/ULBRA)
Teotônio Segurado 1501 SUL – 77054-970 – Palmas – TO – Brasil.
[2]Curso de Pedagogia – Fundação Universidade Federal do Tocantins (UFT)
Av. Lurdes Solino s/nº Setor Universitário – 77650-000 – Miracema – TO – Brasil.

{robertmady,camilagt, thiago27}@gmail.com
nandoluiz@ulbra-to.br,cristianeqm@gmail.com

Abstract

This article shows a web tool, which evaluates and classifies pupils that present characteristcs of attention deficit hyperactivity disorder. The identification of the type of disfuction is important, therefore is necessary to develop specific activities for each group of disorder, thus objectifyingcan, to brighten up the apprenticeship difficulties of this group the pupils, once that these tend to present a deplay in the apprenticeship in relation to the pupils who not present characteristcs of attention deficit hyperactivity. The objective of this tool is to assist in the classification the type of disfunction that possibly compromises the level of attention of the pupils in the lessons, giving chance to the teacher who acts with these pupils, greater understanding concerning the characterization of these children.

Keywords: Disorder, hyperactivity, attention deficit, Bayesian Networks.

Resumo

Este artigo apresenta uma ferramenta web que avalia e classifica os alunos que apresentam características de transtorno por déficit de atenção/hiperatividade. A identificação do tipo de transtorno é importante, pois é necessário desenvolver atividades específicas para cada classe de transtorno, visando assim, amenizar as dificuldades de aprendizagem deste grupo de alunos, uma vez que estes tendem a apresentar um atraso na aprendizagem escolar em relação aos alunos que não apresentam características de déficit de atenção/hiperatividade. A finalidade da ferramenta é auxiliar na classificação do tipo de disfunção que possivelmente compromete o nível de atenção dos alunos nas aulas, oportunizando ao professor que atua com estes alunos, maior compreensão acerca da caracterização destas crianças.

Palavras Chave: Transtorno, Hiperatividade, Déficit de Atenção, Redes *Bayesianas*.

1. INTRODUÇÃO

Embora muitas iniciativas estejam sendo realizadas em busca da melhor compreensão sobre o transtorno por déficit de atenção e hiperatividade, ainda inexistem (ou ainda são pouco divulgadas) pesquisas na área da educação que auxiliem os professores a identificar e usar atividades específicas para cada tipo de transtorno. A história dos primeiros estudos sobre o transtorno por déficit de atenção/hiperatividade confunde-se com a dos distúrbios de aprendizagem, visto que inicialmente os diversos tipos de problemas escolares de base orgânica foram pesquisados como se fizessem parte de um único grupo.

Na década de 60 eram utilizados como sinônimos os termos síndrome do comportamento hipercinético, dislexia, deficiência perceptual, lesão cerebral orgânica, paralisia cerebral mínima, disfunção cerebral mínima e distúrbios de aprendizagem [12]. Desta forma, os problemas de atenção, a hiperatividade e a impulsividade, sintomas atualmente considerados característicos das crianças com transtorno por déficit de atenção/hiperatividade, eram descritos como sendo parte do quadro clínico da "lesão cerebral mínima". Este termo era utilizado para designar transtornos (pré, peri ou pós-natais) no desenvolvimento céfalo-caudal que acarretavam problemas escolares [1].

Para o auxílio de uma possível classificação dos transtornos por déficit de atenção/hiperatividade dos alunos, criou-se uma ferramenta para classificação do transtorno, oportunizando ao professor que atua com um grupo de alunos, maior compreensão acerca da caracterização destes alunos. A identificação e a descrição da classificação do tipo de transtorno se darão a partir dos dados colhidos dos alunos através do uso de um formulário preenchido pelo professor. Esses dados serão analisados por uma rede *bayesiana* (técnica de Inteligência Artificial), criada a partir destes formulários, que mostrará como resultado a probabilidade de existência dos possíveis transtornos.

Esta ferramenta trabalhará com alunos de 1ª à 4ª série do ensino fundamental. A escolha dessas séries se deu porque é nesse período que ocorre o início do aprendizado de cálculos, leitura e escrita, fases estas que se tem mais chances de se identificar algum tipo de transtorno nos alunos.

2. TRANSTORNO POR DÉFICIT DE ATENÇÃO/HIPERATIVIDADE

Na década de 40, o transtorno por déficit de atenção/hiperatividade começou a ser considerado como decorrente de uma "disfunção cerebral mínima", visto que as crianças apresentavam sintomas (hiperatividade, desatenção e desorganização) semelhantes aos daquelas que haviam contraído encefalite letárgica [2]. Os problemas de aprendizagem observados no aluno com "disfunção cerebral mínima" eram considerados como decorrentes de alterações no Sistema Nervoso Central, as quais não afetavam, entretanto, o desenvolvimento intelectual da criança [6].

Por iniciativa da Associação Americana de Psiquiatria [2] o termo "disfunção cerebral mínima" foi substituído, em 1968, por "reação hipercinética" e assim foi apresentado no Manual Diagnóstico e Estatístico de Transtornos Mentais (DSM-II). Doze anos depois, foi proposto no DSM-III [4] o uso do termo "distúrbio por déficit de atenção com ou sem hiperatividade", com a seguinte classificação: "com hiperatividade", "sem hiperatividade" e

"tipo residual". Acreditava-se que tal distúrbio pudesse ser encontrado em 3% na população em idade escolar. Para o diagnóstico diferencial, os sintomas deveriam ter aparecido na criança antes dos sete anos, assim como precisavam estar ocorrendo há pelo menos seis meses [4].

A terminologia empregada no DSM-III foi revisada e novas expressões foram adotadas, em 1987, no DSM-III-R: "transtorno de déficit de atenção com hiperatividade" e "transtorno de déficit de atenção indiferenciado" [5].

A criança do tipo "com hiperatividade" foi caracterizada como sendo hiperativa, impulsiva e com problemas em nível de atenção sustentada [5]. A hiperatividade pode ser identificada através dos seguintes comportamentos, os quais ocorrem freqüentemente:

- Agitação das mãos ou do corpo quando sentada na carteira escolar;
- Dificuldade em permanecer sentada;
- Corrida ou subida em móveis ou objetos em momentos inadequados;
- Dificuldade em brincar silenciosamente;
- Atividade motora excessiva;
- Fala exagerada.

A criança com "déficit de atenção indiferenciado", por sua vez, foi descrita como desatenta e impulsiva, sintomas estes que deveriam persistir até a vida adulta. As características principais de uma criança com déficit de atenção são:

- Falta de atenção a detalhes e realização de erros por descuidos nas tarefas;
- Dificuldade em sustentar a atenção em brincadeiras ou atividades escolares;
- Dificuldade em escutar o outro;
- Dificuldade em seguir instruções e em completar tarefas;
- Dificuldade em organizar atividades;
- Dificuldade em realizar tarefas que exijam esforço mental prolongado;
- Perda de objetos;
- Distração quando da ocorrência de estímulos novos;
- Esquecimento na realização de atividades do dia-a-dia.

O transtorno por déficit de atenção/hiperatividade está sendo estudado há muitas décadas e a terminologia empregada para designar este distúrbio já foi alterada várias vezes. Entretanto, ao longo desse tempo o transtorno continuou sendo definido como um distúrbio do desenvolvimento que se manifesta através de alterações no comportamento. Nos dois tipos de transtornos, os sintomas deveriam ter se iniciado antes dos sete anos, assim como deveriam estar sendo observados há pelo menos seis meses. As principais conseqüências causadas pelo transtorno de déficit de atenção/hiperatividade são:

- Baixo desempenho escolar;
- Dificuldades de relacionamento;
- Sentimento de menos valia;
- Interferência no desenvolvimento educacional e social

Sendo assim, tendo como base os conceitos apresentados nesta seção sobre os transtornos por déficit de atenção/hiperatividade, a próxima seção apresentará a técnica utilizada para auxiliar no processo de identificação.

3 REDE *BAYESIANA*

Para a construção da ferramenta de classificação dos tipos de transtorno por déficit de atenção/hiperatividade, será utilizada a técnica de rede *bayesiana*. Uma rede *bayesiana* consiste em ser uma forma de representar uma probabilidade através do relacionamento entre proposições ou variáveis. Esta técnica de Inteligência Artificial é aplicada sempre que esta relação envolve incerteza ou imprecisão. De acordo com [7], "Redes *bayesianas* constituem um modelo gráfico que representa de forma simples as relações de causalidade das variáveis de um sistema". Uma rede *bayesiana* consiste no seguinte [7]:

- Um conjunto de variáveis e um conjunto de arcos ligados as variáveis.
- Cada variável possui um conjunto limitado de estados mutuamente exclusivos.
- As variáveis e arcos formam um grafo dirigido sem ciclos (DAG)
- Para cada variável A que possui pais B1,...,Bn, existe uma tabela P(A|B1,...,Bn).
- Caso A não possua um pai, a tabela de probabilidades é reduzida para uma probabilidade incondicional P(A).

Ainda segundo [7], uma vez "definida a topologia da rede, basta especificar as probabilidades dos nós que participam em dependências diretas, e utilizar estas para computar as demais probabilidades que se deseje".

> "Geralmente, a rede *bayesiana* é extraída de um especialista que transfere seus conhecimentos de domínio de aplicação para engenheiro de conhecimentos de sistema. Porém, pesquisas vêm sendo realizadas buscando construir redes *bayesianas* utilizando algoritmos capazes de estimar os valores das probabilidades bem como identificar os nós da rede" [9].

Os especialistas, que são pessoas que conhecem sistematicamente os processos ou atividades envolvidas para a realização de uma determinada tarefa, ou conjunto de tarefas, trabalham no mundo real com o fator 'incerteza', devido à inexatidão dos relatos apresentados, e são responsáveis por ponderar estas incertezas e tomar atitudes corretas para cada situação. No ambiente computacional torna-se necessário embutir o fator 'incerteza' nos sistemas desenvolvidos. Para este contexto, a técnica de rede *bayesiana* oferece uma estrutura intuitiva de representar o raciocínio incerto. A vantagem de sua utilização concentra-se no sentido de permitir a representação e manipulação da incerteza com base em princípios matemáticos fundamentados [10].

Os métodos *bayesianos* possibilitam representar numericamente o grau de certeza sobre condições de incerteza, e manipulá-lo de acordo com as regras definidas na teoria da probabilidade, pois a teoria *bayesiana* está fundamentada nesta teoria [3]. De acordo com [11], uma rede *bayesiana* é composta basicamente por duas partes complementares: uma parte qualitativa e outra quantitativa. A parte qualitativa refere-se ao modelo gráfico (grafo acíclico direcionado), onde as variáveis são os nós e as regras, que são as relações de dependência entre as variáveis, são os arcos direcionados. Assim um arco ligado as variáveis A e B, na seguinte forma B→A, indica que a variável A é a conseqüência e a variável B é a causa, e estas apresentam uma relação de dependência, resumidas na regra se B então A.

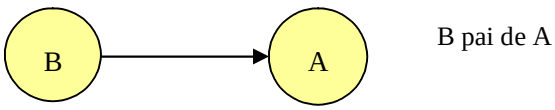


Figura 1: Exemplo de Rede *Bayesiana* – Parte qualitativa

A parte quantitativa de uma rede *bayesiana* interpreta a probabilidade condicional, onde o grau de crença de um agente causa um efeito em outro agente. Assim, P(A|B) reflete a probabilidade de A SE B ocorrer. Considerando P(B) a probabilidade a priori, pois corresponde à probabilidade existente antes de qualquer evidência, e P(A) a probabilidade a posteriori, sendo a probabilidade após conhecer a evidência de B, a regra de *bayes* é representada pelo teorema do mesmo abaixo, na figura 2, que é um método quantitativo para a revisão de probabilidades conhecidas, com base em nova informação amostral [9].

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \mid A)P(A)}{P(B)}$$

Figura 2: Teorema de *Bayes*

A rede *bayesiana* é capaz de realizar dois tipos de cálculos, que são: atualização de crenças e a revisão de crença. Sendo que atualização de crenças resumisse ao cálculo das variáveis aleatórias. E revisão de crença se refere à obtenção das probabilidades das hipóteses diagnosticada e a identificação da hipótese diagnosticada com maior valor de probabilidade [9]. A próxima seção apresentará a ferramenta desenvolvida.

4. APRENDIZ - FERRAMENTA WEB DESENVOLVIDA

O objetivo desta ferramenta desenvolvida consiste em ajudar os professores a detectar em seus alunos a existência ou não de algum tipo transtorno por déficit de atenção/hiperatividade e, a partir daí, auxiliá-lo a conduzir, por meios de atividades pré-definidas, o processo de ensino – aprendizagem.

A ferramenta será implementada e disponibilizada via web para ser acessada por qualquer professor ou escola que deseja utilizá-la. Os professores poderão cadastrar seus alunos no sistema e, em seguida, preencher um formulário para analisar se o aluno possui ou não algum distúrbio de atenção. Com essa ferramenta o trabalho do professor torna-se mais rápido e eficiente, devido o fato dele ganhar tempo na execução das atividades realizadas em sala de aula. Nas próximas seções serão explicadas, com mais detalhes, algumas particularidades dessa ferramenta.

4.1 Obtenção dos Dados – Modelo de Formulário da Aplicação

Para obter os dados que serão submetidos para a análise da rede *bayesiana*, assim como para sua atualização, serão utilizados os formulários propostos por [8]. Estes formulários serão disponibilizados aos professores, o qual será o responsável por seu preenchimento. Desta forma, será possível detectar o tipo de transtorno existente na criança, pois cada formulário é composto por um conjunto de variáveis que auxiliam na descoberta do distúrbio.

Para cada tipo de transtorno é aplicado um formulário diferente, onde cada formulário estará dividido em três grupos, e cada grupo irá corresponder a um determinado transtorno. Esses grupos estão divididos em grupo I, II e III, onde:

- Grupo I: contém um conjunto de variáveis pertinentes à descoberta do transtorno "déficit de atenção", onde o formulário conterá dezessete perguntas a serem preenchidas, referentes a nove itens descritos no DSM-IVR como características do déficit de atenção tipo predominantemente desatento, desses nove itens distribuídos nas dezessete questões,

o aluno deverá corresponder no mínimo a seis questões para poder chegar à conclusão de que realmente o aluno possua o "déficit de atenção".

- Grupo II: este irá conter seis perguntas referentes ao déficit de atenção/hiperatividade, predominantemente "hiperativo", onde dessas seis perguntas, o professor deverá responder pelo menos três questões para se concluir que o aluno apresente características de hiperatividade.
- Grupo III: Esse grupo foi organizado em três questões; e para o aluno ser caracterizado como predominantemente hiperativo/impulsivo, deve responder a pelo menos três questões do grupo II e três questões do grupo III.
- Outra variação do transtorno é o tipo combinado. Para detectar se o aluno possui os dois tipos de transtornos (déficit de atenção/hiperatividade) é necessário que o professor responda pelo menos seis das questões do grupo I, mais três do grupo II e mais as três últimas perguntas do grupo III, totalizando um total de doze perguntas a serem respondidas, observando que as três últimas questões do grupo III devem estar devidamente preenchidas, pois elas são variáveis de extrema importância que contribuirão na conclusão desses dos dois tipos de transtornos, por identificar no aluno o comportamento impulsivo.

A seguir (Tabela 1) será apresentado parte do modelo de questionário para a avaliação do tipo de transtornos por déficit de atenção.

Tabela 1: Parte do formulário modelo para analisar Déficit de Atenção/Hiperatividade [8].

GRUPO I				
1) A criança deixa de prestar atenção a detalhes no decorrer das atividades escolares?	Sempre □	Nunca □	□ Às vezes. ____ vezes por semana. □ Dependendo da tarefa. Qual?_____	□ Não tenho essa informação
2) A criança comete erros por descuido (desleixo) nas atividades escolares ou brincadeiras?	Sempre □	Nunca □	□ Às vezes. ____ vezes por semana. □Dependendo da tarefa ou brincadeira. Qua2l?_____	□ Não tenho essa informação
3) A criança tem dificuldade em manter a atenção nas atividades escolares?	Sempre □	Nunca □	□ Às vezes. ____ vezes por semana. □ Dependendo da tarefa. Qual?_____	□ Não tenho essa informação
4) A criança tem dificuldade em manter a atenção nas brincadeiras?	Sempre □	Nunca □	□ Às vezes. ____ vezes por semana. □ Dependendo da brincadeira. Qual?_____	□ Não tenho essa informação
5) A criança parece não escutar quando o colega ou o professor lhe dirigem a palavra?	Sempre □	Nunca □	□ Às vezes. ____ vezes por semana. □ Dependendo da situação. Qual?_____	□ Não tenho essa informação
6) A criança não consegue seguir as instruções dadas pelo professor?	Sempre □	Nunca □	□ Às vezes. ____ vezes por semana. □ Dependendo da tarefa. Qual?_____ Por que?_____	□ Não tenho essa informação

4.2 Processamento – Modelo de Rede *Bayesiana*

Para o desenvolvimento do modelo da rede *bayesiana* foi necessário à ajuda de um especialista do domínio, conhecedor da forma de caracterização e classificação dos

transtornos envolvidos. Com o auxílio deste especialista puderam-se ponderar as incertezas ou falhas na obtenção dos dados colhidos nos formulários e assim agir de forma correta para cada situação combinada. Com o possível preenchimento errôneo dos formulários por parte dos professores o fator incerteza é uma variável que poderia surgir em qualquer momento, acarretando assim uma classificação errada do transtorno. Porém, com a escolha da utilização da técnica de rede *bayesiana*, têm-se a possibilidade de representar e manipular esses erros ou incertezas de forma estruturada e baseada em fundamentos matemáticos.

A ferramenta disponibilizará aos professores um formulário para a obtenção dos dados dos alunos. Após obter as informações necessárias, a ferramenta submete estes dados para serem analisados pela rede *bayesiana*. Assim a ferramenta desenvolvida, será expandida a totalidade de nós que são necessários para a identificação do tipo de transtorno.

O sistema tem em sua base de conhecimento fatos e regras que representam o perfil das crianças, colhidos a partir dos formulários aplicados ao professor responsável. Os fatos e às regras são associadas às incertezas presentes no domínio, e é explicitada a crença em sua ocorrência através de valores de probabilidade. O raciocínio realizado pelo sistema deve considerar estas probabilidades para que, a partir dos dados de entrada associar um vetor de probabilidades ao conjunto de hipóteses classificatórias.

A parte qualitativa é composta por nove variáveis (questões do formulário), podendo ser nove variáveis de hiperatividade ou de déficit de atenção, dependendo do formulário a ser preenchido pelo professor, onde a variável central é conseqüência (o transtorno) de todas as opções de nós das outras variáveis. Ou seja, os nós periféricos são as causas, responsável pelos índices probabilísticos do nó central, cada opção escolhida nos nós periféricos causa conseqüências nos resultados obtidos no nó central. O modelo gráfico ilustrado na Figura 3 descreve a parte qualitativa para a rede *bayesiana* do transtorno hiperatividade.

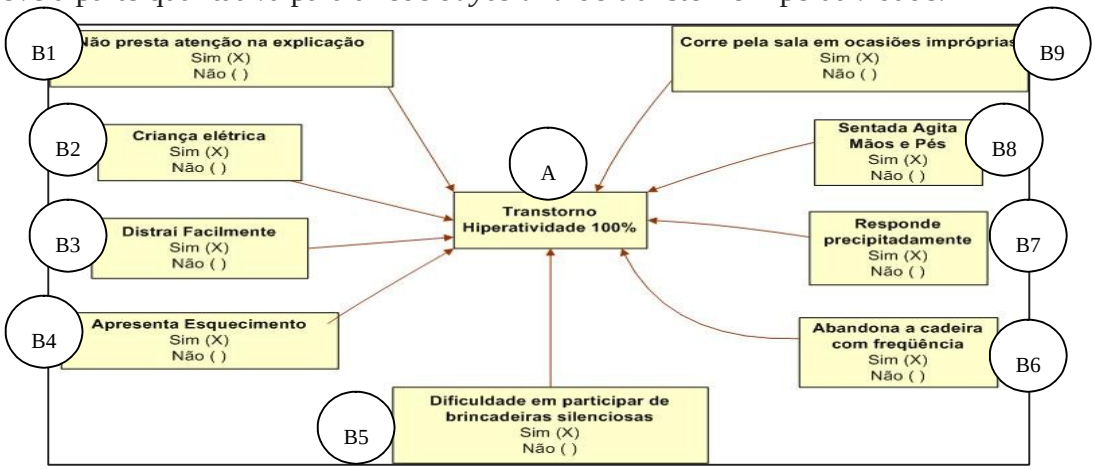


Figura 3: Rede *bayesiana* - Parte Qualitativa

A parte quantitativa surge, a partir da probabilidade da escolhas de cada nó em suas respectivas variáveis, onde um nó reflete uma probabilidade diferente para cada combinação de nó escolhido em cada variável. A Figura 4 descreve a parte quantitativa para a rede *bayesiana* do transtorno hiperatividade, onde V representa as variáveis marcadas com Sim e F as variáveis marcadas com Não.

P(A | B1^B2 ^B3^B4^B5^B6^B7^B8^B9)

(Hiperatividade)									Transtornos em (%).
B1	B2	B3	B4	B5	B6	B7	B8	B9	A
V	V	V	V	V	V	V	V	V	100
...	...	...	...	...	...	...	...	...	...
V	V	V	V	V	F	F	F	F	83
...	...	...	...	...	...	...	...	...	...
V	V	V	V	F	F	F	F	F	67
...	...	...	...	...	...	...	...	...	...

Figura 4: Rede *bayesiana* - Parte Quantitativa (baseado no teorema de *bayes*).

Aos fatos e as regras são associadas às incertezas presentes no domínio através de valores de probabilidades. O raciocínio realizado pelo sistema deve considerar estas probabilidades iniciais (probabilidades a priori) obtidas da pesquisa, para que o sistema, a partir dos dados de entrada (evidências), possa calcular a classificação mais provável (probabilidades a posteriori).

4.3 Apresentação dos Resultados

Os formulários serão aplicados aos professores, que são os responsáveis por preencherem as questões impostas para cada aluno que tem relação com o mesmo. Algumas das questões, referente aos formulários para obtenção dos dados, são apresentadas na Figura 5.

Figura 5 e 6: Tela onde é aplicado o formulário para classificação do transtorno e Apresentação do resultado da classificação do transtorno.

Primeiramente os professores, escolhem o formulário correspondente ao tipo de transtorno selecionado, em seguida, preenche o formulário de acordo com as características de uma possível disfunção que o aluno vem a ter, representado na figura 5. Lembrando que para o professor responder o questionário, é preciso que ele tenha no mínimo seis meses de convivência com o aluno, isso se deve ao fato de que o professor necessita primeiro conhecer o aluno e ter esse convívio durante esse tempo, para depois interpretar o comportamento do aluno em sala de aula. Por fim, o professor depois de preencher as questões do formulário poderá classificar o transtorno do aluno correspondente através do botão "Classificar Transtorno", que por sua vez vai analisar as respostas preenchidas no formulário e em seguida, será carregada uma rede *bayesiana*, onde irá processar o resultado do questionário e apresentar o resultado do possível distúrbio do aluno em porcentagem.

Após o preenchimento do formulário é gerado o resultado, ou melhor, é classificado o transtorno que o aluno possivelmente possui. Conforme a Figura 6, a classificação dos transtornos é dada em porcentagem. Após ser mostrada a porcentagem de cada transtorno, de acordo com o transtorno predominante é mostrada as causas do transtorno e suas possíveis soluções para auxiliar o professor a trabalhar melhor com esses alunos que possui esse distúrbio.

É importante ressaltar que essa ferramenta auxilia o professor a trabalhar com alunos que possui dificuldade na aprendizagem e não conseguem fixar sua atenção por muito tempo em uma única atividade. Dessa forma, é necessário trabalhar com exercícios específicos para ajudar a suprir a deficiência que cada aluno possui, sendo assim, a ferramenta propõe ao professor trabalhar com materiais que buscam prender a atenção do aluno em uma determinada atividade, ou materiais que auxiliem o aluno na compreensão das fórmulas matemáticas, na escrita e na leitura, que são justamente nessas atividades que o aluno sente a dificuldade. O intuito da ferramenta não é diagnosticar e afirmar 100% de certeza que o aluno possui ou não algum tipo de distúrbio, pois isso só pode ser feito por um profissional da área, que no caso é um psicólogo, mas o objetivo da ferramenta é justamente auxiliar o professor a conduzir melhor, seus alunos em sala de aula, não substituindo o profissional que é o verdadeiro responsável pelo acompanhamento da deficiência desses alunos.

5. CONSIDERAÇÕES FINAIS

A ferramenta tem por base auxiliar os professores na detecção de transtornos por déficit de atenção/hiperatividade de seus alunos, mas não terá somente o cuidado de classificar os possíveis transtornos, pois ela também terá a responsabilidade de mostrar as causas para a existência dos mesmos e suas possíveis soluções. Estas soluções poderão ser aplicadas pelos professores a partir do momento em que for detectado o tipo de transtorno que o aluno sofre e assim aplicar medidas que possibilitarão trabalhar em pontos específicos do distúrbio, melhorando dessa forma o nível de aprendizado dessa criança.

A ferramenta será composta, a princípio de dois módulos, um voltado para o professor, o qual foi apresentado neste artigo, e o outro voltado para o aluno, que ainda será desenvolvido, oferecendo ferramentas para auxiliar o professor no processo de ensino e de aprendizagem. O primeiro módulo contará com a parte gerencial da ferramenta, ou seja, cadastros de alunos, professores, escolas, etc, além da classificação dos transtornos através da aplicação dos formulários. Este módulo já está implementado e contou com a ajuda de um especialista no domínio para montagem das redes *bayesianas*, as quais foram baseadas nos formulários do

trabalho de mestrado de [8]. Já o segundo módulo contará com técnicas de iteração com os alunos que apresentam algum tipo de déficit abordado. Estas técnicas abordarão desde jogos de computadores direcionados a área educacional e que estimulam o desenvolvimento matemático, ortográfico, léxico, visual através de imagens, som, interação com textos, até abordagem simples com textos. Esse módulo necessitará de um melhor aprofundamento no estudo de computação gráfica e por esse motivo ficará pra ser desenvolvido futuramente, pois é de fundamental importância ter essa ferramenta auxiliando professores no processo de desenvolvimento dos alunos que apresentam algum tipo de transtorno por déficit de atenção/hiperatividade.

6. REFERÊNCIAS BIBLIOGRÁFICAS

[1] CRUICKSHANK, W.N.; JOHNSON, J.O. **A educação da criança e do jovem excepcional.** Porto Alegre: Globo, 1979. v.1.

[2] ERK, R.R. The evolution of attention deficit disorders terminology. In: **Elementary School Guidance & Counseling,** v.29, n.4, p.243-248, 1995.

[3] HECKERMAN, David. A bayesian approach to Learning Causal Networks. Technical Report Msr-Tr-95-04. Microsoft Research, (1995). Disponível em:< **http://research.microsoft.com/research/pubs/view.aspx?tr_id=65** >. Acesso em: 15/05/2007.

[4] KAPLAN, H.I.; SADOCK, B.J. **Compêndio de psiquiatria dinâmica.** Porto Alegre: Artes Médicas, 1984.

[5] KAPLAN, H.I. et al. **Compêndio de psiquiatria.** Porto Alegre: Artes Médicas, 1990

[6] LEFÈVRE, A.B.; LEFÈVRE, B.H. **Disfunção cerebral mínima:** estudo multidisciplinar. São Paulo: Sarvier, 1983.

[7] MARQUES, Roberto Ligeiro, DULTRA, Inês. Redes Bayesianas: o que são, para que servem, algoritmos e exemplos de aplicações. Disponível em: **http://www.cos.ufrj.br/~ines/courses/cos740/leila/cos740/Bayesianas.pdf.** Acessado em: 25/05/2007.

[8] MANSANERA, Cristiane de Quadros. **Transtorno por déficit de atenção/hiperatividade e problemas de aprendizagem.** Maringá, Paraná: Universidade Estadual de Maringá, 2002.

[9] NASSAR, S. M. (2003), **Tratamento de Incerteza:** Sistemas Especialistas Probabilísticos. Pós-Graduação em Ciências da Computação: Centro Tecnológico. Departamento de Informática e Estatística. Universidade Federal de Santa Catarina.

[10] OLIVEIRA, Fernando Luiz de, Bet, Sabrina. **"Modelo de Interfaces Adaptativas utilizando Redes Bayesianas"**, Palmas – TO, 2003.

[11] PEARL, Judea. RUSSELL, Stuart. **Bayesian Networks**, Disponível em <http://ftp.cs.ucla.edu/pub/stat_ser/R277.pdf >. Acessado em 02 de junho de 2007.

[12] TARNOPOL, L. Introdução aos distúrbios neuropsicológicos de aprendizagem. In: **Crianças com distúrbios de aprendizagem:** diagnóstico, medicação e educação. São Paulo, EDUART, 1980. p. 15-31.

Fundamentos para el Submódulo Evaluador en Sistemas Tutores Inteligentes: Diagnóstico, predicción y autoevaluación

Zulma Cataldi,
LIEMA - Laboratorio de Informática Educativa y Medios Audiovisuales. Facultad de Ingeniería. UBA.
Facultad Regional Buenos Aires. Universidad Tecnológica Nacional
liema@fi.uba.ar

Fernando Salgueiro,
LIEMA - Laboratorio de Informática Educativa y Medios Audiovisuales. Facultad de Ingeniería. UBA.
fsalgueiro@fi.uba.ar

Fernando Javier Lage[1]
LIEMA - Laboratorio de Informática Educativa y Medios Audiovisuales. Facultad de Ingeniería. UBA.
Facultad Regional Buenos Aires. Universidad Tecnológica Nacional.
flage@fi.uba.ar

Abstract

In this paper, we present the basic functions of the evaluator submodule and how it will carry out diagnosis and evaluation. Taking into consideration the mistakes committed in the evaluations and its analysis, a diagnosis is carried out. Based on data of the first evaluations, the prediction of future mistakes is useful to give suggestions, bearing in mind improving performance by making a corrective exercitation. In order to carry out this prediction of future errors, a neuronal network is used and based on this; a sequence of exercises and problems is suggested to the student in order to improve his production and therefore his grades.
Regarding the function of the evaluation, in order to help in the process of new conceptualization of errors, an application has been made so that students can carry out their own auto evaluation. Different series of questions based upon the errors were elaborated; each of them corresponds with a Didactic Unit already studied. This instrumentation intends to make the student realize his weaknesses and therefore, would be able to reinforce them before his partial or final evaluation.

Key words: *Intelligent Tutoring Systems, Neural networks, Evaluation, Prediction*

Resumen

Se presentan las funciones básicas del submódulo evaluador y se describe cómo se llevará cabo el diagnóstico y la evaluación. Con base en los errores cometidos en sus evaluaciones y a partir del análisis de los mismos se busca efectuar un diagnóstico. En función de los datos de las primeras evaluaciones se podrán predecir los errores futuros para sugerirle ejercitación correctiva a fin de mejorar el rendimiento. Para efectuar la predicción de los errores siguientes se usa una red neuronal y sobre esta base se le sugiere al estudiante una secuencia de ejercicios y problemas a fin de mejorar su producción y por lo tanto sus calificaciones.
En cuanto a la función de evaluación a fin de ayudar en el proceso de reconceptualización de los errores, se ha realizado una aplicación para que los estudiantes puedan efectuar su autoevaluación. Se han elaborado diferentes series de preguntas con base en los errores que se corresponden a cada Unidad Didáctica trabajada. Con esta instrumentación se busca que el alumno pueda *darse cuenta* de sus debilidades y pueda reforzarlas antes de su evaluación parcial o final.

Palabras Clave: *Sistemas Tutores Inteligentes, Redes neuronales, Evaluación, Predicción.*

[1] Esta comunicación ha sido desarrollada en el marco del Convenio FI-UBA y UTN-FRBA. Proyecto *Modelado del tutor basado en redes neuronales para un Sistema Tutor Inteligente* del Programa de Incentivos 2007-2008.

1. INTRODUCCIÓN

En publicaciones previas [1] se ha señalado, el bajo rendimiento de los estudiantes de Programación Básica[2] en sus evaluaciones parciales y finales durante los últimos seis cuatrimestres [1]. Este trabajo de investigación surge buscando una solución a este problema a través de la predicción tomando como datos los resultados en las evaluaciones y su análisis para poder efectuar luego un diagnóstico, a fin de sugerir estrategias de refuerzo, ya que se piensa que un sistema predictor podría asistir a los estudiantes en su trayecto [2]. En la actualidad no existen sistemas expertos que resuelvan en forma eficiente este problema, ya que sólo se encuentran disponibles algunos programas para enseñanza de programación, pero de tipo tutorial [3,4].

La autoevaluación se fundamenta sobre bases epistemológicas provenientes de las corrientes que abogan por la autorregulación del alumno y se justifica a partir del problema del desgranamiento sostenido en los cursos iniciales de Programación Básica en la universidad. Se creó una herramienta a partir del listado de los errores que los alumnos comenten en las evaluaciones parciales y finales. Éstos fueron registrados y clasificados conformando una base de datos refinada para poder componer la autoevaluación diseñada en base a ejercicios para reforzar los aspectos críticos.

2. DESARROLLO

El sistema planteado resolvería el problema de la predicción del comportamiento de los alumnos en el marco universitario, dado que a partir de los primeros exámenes se podrían predecir posibles errores futuros a través del diagnóstico y la predicción, a fin de sugerir la ejercitación correctiva a para mejorar su rendimiento y encauzar su aprendizaje hacia conceptualizaciones incorporadas de forma más significativa y permanente [5]. Los errores cometidos por cada alumno en los exámenes son los datos de partida, se cargan en una base de datos y permiten efectuar una primera categorización en grandes grupos de temas problemáticos. A partir de estos datos el sistema que se propone deberá ser capaz de predecir las próximas fallas de un alumno y como consecuencia de ello, deberá determinar la siguiente secuencia de problemas y/o ejercicios que el estudiante deberá realizar para mejorar su rendimiento y por ende sus calificaciones [6].

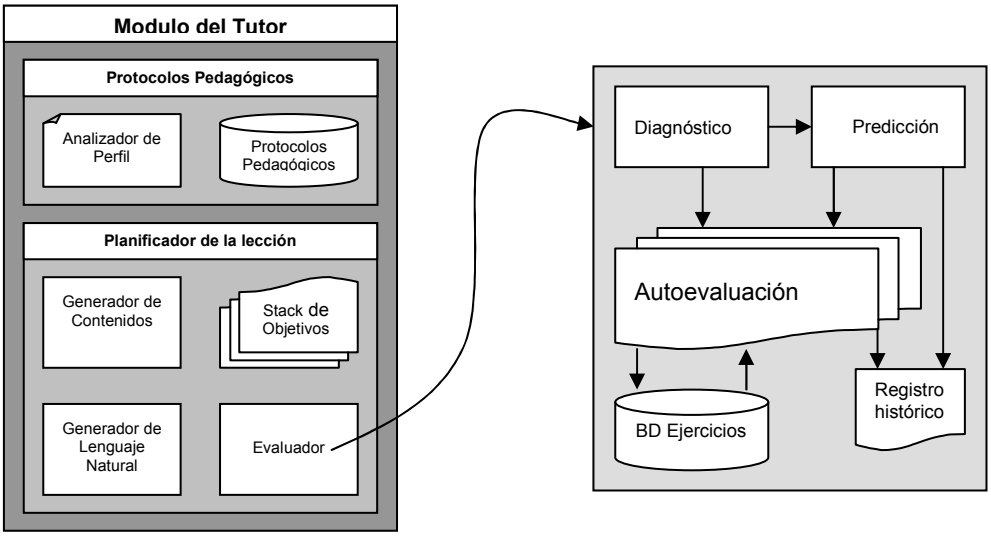


Figura 1: Módulo del tutor y submódulo Evaluador

En la Figura 1 se observa el esquema del módulo tutor y el submódulo evaluador con sus funciones básicas.

[2] En Facultad de Ingeniería se corresponde a las asignaturas: Computación que se dicta para todas las carreras excepto Informática y Electrónica y Algoritmos y Programación I para Ingeniería Informática y Licenciatura en Análisis de Sistemas.

El problema del diagnóstico y la predicción incluye:
- una primera etapa de diagnóstico, que está determinada por las respuestas dadas por los alumnos en sus exámenes parciales y finales,
- una segunda parte basada en la predicción de los errores futuros y en la determinación de la ejercitación que deberá resolver un alumno a fin de mejorar su rendimiento.

Este problema está relacionado directamente con los contenidos de la asignatura y además debe considerar, desde el punto de vista didáctico, las acciones del tutor y del estudiante que conforman el triángulo didáctico, en tanto es un proceso comunicacional [4]. Por lo tanto debería plantearse una solución con base en los sistemas inteligentes [7,8,9] buscando predecir comportamientos futuros sugiriendo líneas de acción, tomando alguna decisión pedagógica [9,10].
En un STI (Sistema Tutor Inteligente) existe un *modelo de dominio* donde se encuentra el conocimiento sobre el dominio en particular, de tipo declarativo (los primeros principios, la comprensión del dominio) y procedural (el conocimiento que es utilizado para realizar una tarea) [11]. Este conocimiento del dominio consiste en los hechos y en las relaciones entre ellos que, generalmente debe ser fortalecido en general por uno o más especialistas [12,13]. Este módulo de dominio, tiene algunas funciones básicas, ya que sirve como fuente del conocimiento a ser presentado al estudiante, lo que incluye la generación del material, generación de preguntas y de respuestas que conforma un patrón que permitirá evaluar el conocimiento del estudiante. Para ello, el sistema debe ser capaz de generar soluciones a los problemas en el contexto del estudiante, para que sus respuestas puedan ser evaluadas. La base del conocimiento del dominio es un componente clave en el *sistema predictor,* ya que es ahí donde está representado el material de enseñanza [14,15]. En los casos en que el dominio sea de naturaleza descriptiva y teórica (como en geografía ó física), la representación utilizada es la declarativa a través de redes semánticas ó *"frames"*. En los casos en que el dominio esté orientado a una tarea, tal el caso de la programación, la representación tiende a ser procedural, ya que son originalmente reglas de producción [16]. El modelo del dominio es un tema que ha sido estudiado[15, 16], así como el modo en que el sistema lo puede usar para razonar. Algunas de las representaciones posibles son las redes semánticas, reglas de producción y "*constraints*" [15, 16]. Su elección depende de cómo se lo usará ya que es común a todos los usuarios del sistema. A fin de dar solución a la problemática planteada de predicción y diagnóstico, se indagará sobre la aplicación de redes neuronales que han dado buenos resultados en diversas áreas [17].

También se hará un acercamiento a la problemática desde los aportes que puede hacer *la autoevaluación* a la formación del estudiante, en la adquisición de experticia en el dominio de la resolución de problemas en ingeniería desde los contenidos de la Programación Básica. En este sentido:

- Se propondrá una herramienta para que el estudiante pueda descubrir e interpretar cómo puede hacer para resolver correctamente sus problemas trabajando desde la reconceptualización de sus errores.

3.1. DIAGNÓSTICO Y PREDICCIÓN

3.1.1. Las redes neuronales

Las redes neuronales (RN) son conjunto de elementos más simples que se interconectan en paralelo en forma jerárquica y que interactúan como los sistemas neuronales psicológicos [18]. A fin de poder utilizarlas para representar sistemas de mayor complejidad pueden tener retroalimentación. Una de sus características diferenciales es que pueden aprender de la experiencia a través de la generalización de casos [19]. Una red neuronal se caracteriza por cuatro elementos básicos: *su topología, el mecanismo de aprendizaje, tipo de asociación realizada entre la información de entrada y salida y la forma de representación de estas informaciones.*

Las neuronas se distribuyen en la red formando capas de un número determinado de elementos básico. Es decir, existe una capa de: *entrada* que recibe directamente la información proveniente de las fuentes externas de la red, capas *ocultas* que son internas a la red y no tienen contacto directo con el exterior (desde cero niveles hasta un número elevado), pudiendo estar interconectadas de distintas maneras, lo que determina junto a su número, las distintas topologías y una capa de *salida* que transfiere la información de la red hacia el exterior. La topología de las redes neuronales es la forma de organización de las neuronas en la red formando capas o agrupaciones de neuronas más ó menos alejadas de la entrada y la salida de la red. Por lo tanto, los parámetros fundamentales de la red serán: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones ente neuronas.

3.1.2. El algoritmo backpropagation

Rumelhart, Hinton y Williams [20] desarrollaron un método de aprendizaje automático que permitió que una red neuronal basada en el *perceptron* [21] aprendiera la asociación existente entre los patrones de entrada y las salidas correspondientes. El *Aprendizaje Automático (Machine Learning)* se puede definir como un conjunto de programas computacionales que mejoran con la experiencia y son sistemas que deben ser capaces de aprender a reconocer patrones a partir de la resolución de problemas o mediante ejemplos provistos por un instructor ó supervisor. Luego, se modificó el *perceptron* de Rosenblatt [21] agregándole capas ocultas, con conexión hacia adelante y sin conexiones recurrentes [20]. Pero, no fue suficiente con introducir algunas modificaciones topológicas a la red, sino que se requerían modificaciones en el algoritmo de aprendizaje; por lo tanto fue desarrollado el método de aprendizaje supervisado basado en la regla *delta generalizada* [20], logrando así, una ampliación del rango de aplicación de las redes neuronales.

El funcionamiento general de una red neuronal de este tipo, se puede dividir en dos partes: *una etapa de entrenamiento y una etapa de puesta en marcha.* La primera consiste en el aprendizaje de un conjunto predefinido de observaciones de entrada–salida dados como ejemplo (utilizando *n* atributos de entrada y un único atributo ó clase, de salida), empleando un ciclo propagación-adaptación de dos fases [17,22], donde en la *primera fase* se aplican los atributos de entrada a la capa de entrada de datos a la red y los valores generados se propagan desde esta capa hacia las superiores hasta generar una salida, en la capa de salida de la red. Para realizar el entrenamiento, se compara el resultado obtenido en cada neurona de salida con el valor deseado para cada neurona en particular y obteniéndose un error para cada una de las unidades de salida. En la *segunda fase*, los errores de las unidades de salida se transmiten hacia atrás, pasando por todas las neuronas de las capas intermedias que contribuyan directamente a la salida, recibiendo el porcentaje de error aproximado a la participación de las neuronas intermedias en la salida original. Este proceso se repite capa por capa hasta llegar a la capa de entrada y hasta que cada neurona haya recibido un error que describa su aporte al error total. Debido a ello, el algoritmo se denomina también de *retro-propagación* o propagación hacia atrás. Los errores se calculan con respecto a los aportes de las neuronas desde la capa de salida hasta la capa de entrada y se reajustan los pesos de las conexiones entre cada par de neuronas en la red, de manera tal que el error total cometido para ese patrón disminuya con respecto al valor del error recibido.

Dado que la fase de funcionamiento es similar a otras redes neuronales artificiales se debe realizar un análisis más profundo del método de aprendizaje. El método de *backpropagation* utiliza una función o superficie de error asociada a la red, buscando un estado de mínimo error estable a través del camino descendente sobre la superficie de error [20]. Es por esto que se debe realizar la retroalimentación para realizar las modificaciones en los pesos iniciales en un valor proporcional al gradiente decreciente de dicha función de error. En la Figura 21 se puede ver un esquema de este tipo de redes neuronales artificiales.

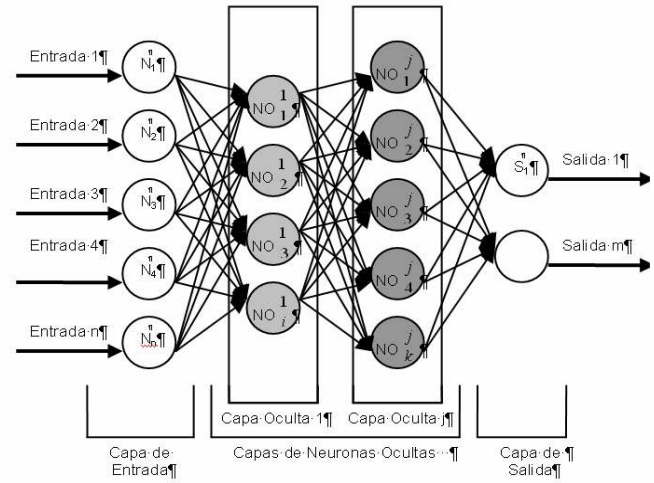


Figura 2: Modelo de red neuronal backpropagation [20].

En la etapa de aprendizaje, se busca minimizar el error entre la salida obtenida por la red y la salida deseada luego del entrenamiento con el conjunto de datos patrones. Es por ello, que en estas redes el aprendizaje es de tipo supervisado, ya que es el usuario o supervisor quien determina cuál es la salida deseada ante la presentación de un patrón de entrada dado

Para modificar los pesos se sigue la fundamentación matemática del algoritmo basado la técnica del *gradiente decreciente* [20].

La función de activación utilizada en cada una de las neuronas debe ser derivable de primer orden.

Para obtener una aproximación funcional óptima se deben elegir cuidadosamente las variables a emplear, es decir se trata de incluir en el modelo las variables que realmente predigan la variable dependiente o de salida, pero que no covaríen entre sí [23], debido a que podrían provocar un *sobreajuste* (*overfiting*) innecesario. Esto sucede cuando el número de parámetros o de pesos de la red resulta excesivo en relación al problema a tratar y al número de patrones de entrenamiento disponibles. El *sobreajuste* disminuye la capacidad de la red de proporcionar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento. Se entiende por generalización de la red a la capacidad de dar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento [17].

3.2. La autoevaluación

En las evaluaciones parciales y finales *es donde aparecen las fallas documentadas a través de los errores que son registrados. Si bien su valor como indicadores de aprendizajes es relativo pues hay que considerar que la misma, es una situación de alto estrés donde el alumno esta más propenso a cometer fallas, es importante su recurrencia.*

Se construyó un registro de los errores cometidos en las evaluaciones parciales y finales que se plasmó en un listado, donde se observan los diferentes tipos de fallas cometidas en forma reiterada por los estudiantes. Ante sus fallas *e*n las etapas iniciales del curso, el estudiante pierde más fácilmente la motivación y desiste de seguir cursando. Los errores citados son: *a) No leen detenidamente el problema a resolver, a veces no saben leer e interpretar el enunciado e interpretan lo que ellos quieren interpretar. b) Al principio confunden Read con Write c) No entienden para qué definir tipos y los definen mal d) Confunden los signos > y < e) Confunden variables y constantes f) No definen constantes g) Confunden filas y columnas con lo que se les dificulta el trabajo con vectores y matrices. h) Manejan mal los subíndices. i) No interpretan las estructuras cíclicas ni sus diferencias j) Usan if anidados en lugar de and k) No inicializan variables y contadores. l) No entienden cómo establecer sangrías por bloques y lo hacen mal m) No diferencian división real y entera n) No discriminan variables dependientes e independientes. ñ) No validan los datos de entrada a los programas o) No documentan los programas p) No verifican la división por cero, de haber un cociente q) Construyen ciclos infinitos*

A su vez, estos errores se pueden agrupar más globalmente de acuerdo a la clasificación siguiente: *a) Errores debidos a la mala interpretación del problema a resolver. b) Errores diversos en el uso del lenguaje c) Errores debido a procesos de decisión mal elaborados.*

La dificultad en la comprensión de los procesos de enseñanza y de aprendizaje se debe en parte a la multiplicidad de variables involucradas en ambos. Dentro de estos procesos, la evaluación da cuenta

de los errores cometidos en forma reiterada y representa uno de los aspectos de más trascendencia en el acto didáctico. Es frecuente, observar los diferentes tipos de errores cometidos por los estudiantes en las evaluaciones, que van desde omisiones leves debidas a la falta de atención o debidos al nivel estrés del proceso evaluativo, hasta errores conceptuales graves que dan lugar a la reprobación. La literatura existente determina el proceso de evaluación desde los aspectos cuantitativos de los errores, y aún en aquellos casos en que se realiza una consideración cualitativa, esta se lleva a cabo con un criterio globalizador [26].

Uno de los núcleos conceptuales que conforman la columna vertebral de la didáctica es la evaluación. Su importancia y necesidad ha sido puesta de manifiesto por numerosos autores e investigadores de la educación como Hernández Rojas [27] y House [28]. Angulo y Blanco [29] dicen al respecto: *"la evaluación es el proceso por el cual conocemos y valoramos la calidad del servicio y el papel de los distintos componentes en el mismo"*; significa entonces, que evaluar implica comprender qué le sucede al alumno y a todos los integrantes implicados en el proceso de enseñanza y aprendizaje considerado.
Como el *estudiante es el principal protagonista* y destinatario del proceso de enseñanza y aprendizaje, de algún modo se lo puede hacer partícipe de su evaluación a través de *la autoevaluación.* Blanco Prieto [30] dice: *"la autoevaluación por parte del alumno supone una concepción democrática y formativa del proceso educativo"* y Gimeno Sacristán [31], al respecto, enumera, entre otras, una serie de razones que justifican la necesidad de la *autoevaluación*: a) una persona formada adecuadamente es aquella capaz de dirigir su propio aprendizaje; por lo tanto habrá que preparar al estudiante para que participe en su evaluación, b) cuando un estudiante es capaz de corregirse en un determinado contenido, se puede decir que domina ese contenido, c) con la autoevaluación los alumnos se sentirán tratados como adultos responsables y d) la importancia de este tipo de proceso evaluativo radica en que es una actividad que mejora los aprendizajes de los alumnos, que en definitiva, es el fin que se persigue.
Retomando la idea de que el estudiante debe ser partícipe de su proceso de aprendizaje, y adscribiendo a la justificación de Gimeno Sacristán [31] es posible fundamentar la autoevaluación en el *aprendizaje autorregulado.* Puede decirse que existe aprendizaje autorregulado, *cuando el alumno es capaz de ser un participante activo de su proceso desde el punto de vista metacognitivo, conductual y motivacional.* [32,33]. Esto implica que él debe lograr una *toma de conciencia* de sus propias dificultades para establecer estrategias de acción (aspecto *metacognitivo*) y por otra parte debe ser capaz de controlar su conducta para alcanzar sus metas (aspecto *conductual* y *motivacional*).

4. OBJETIVOS DEL TRABAJO

El objetivo general se desglosó en tres objetivos específicos:

- Determinar el tipo de red neuronal a usar para determinar en forma más eficiente la asesoría luego del diagnóstico realizado.
- Aplicar redes neuronales para predecir errores en futuros exámenes y a partir de éstos determinar que clase de problemas o ejercicios debe resolver el alumno para mejorar su rendimiento, en un dado dominio.
- Evaluar el sistema utilizando datos estadísticos almacenados y con un grupo de docentes y de estudiantes disponibles.
- Proponer y poner a prueba una herramienta que permite llevar acabo la autoevaluación.

5. METODOLOGÍA

1) Se determinó la herramienta inteligente más apropiada para resolver la cuestión de la predicción en el comportamiento del alumno y la determinación de la actividad futura que deberá realizar, entre las disponibles en un dado dominio.

2) Se creó una base de datos con todos los ejercicios tomados en las evaluaciones parciales y finales de una Programación Básica, indicando su tipo (ya sea de tipo práctico o teórico), su dificultad, su nivel de obligatoriedad en una evaluación, su puntaje y su objetivo de aprendizaje. El tamaño de la muestra fue de 450 alumnos, usándose 2/3 para entrenamiento y 1/3 para pruebas.
3) Se elaboraron las bases para la creación de un sistema predictor que cumpla con los objetivos propuestos. Para este punto se determinarán las herramientas metodológicas más apropiadas que provee la ingeniería de software y los sistemas inteligentes en cuanto a métodos, técnicas y herramientas disponibles.
4) Se realizó una prueba piloto (evaluación interna y externa del software) con un grupo estudiantes y docentes de la cátedra disponibles.
5) Se diseño el sistema evaluador desde la propia autoevaluación y se probó con alumnos disponibles de cursos de Programación Básica

6. PARTE EXPERIMENTAL APLICANDO LA RED NEURONAL

6.1. Descripción

Las etapas seguidas en la parte experimental fueron las que se describen en la Tabla 1. Se utilizó la herramienta *NNclass* [25]. La misma posee un instructivo que permite efectuar el ingreso de los datos. El valor máximo es de 50 variables de entrada y 40 de ellas pueden de tipo categóricas.

Please make sure that there are no more than 50 neurons in Input Layer.
There should be exactly 1 Output variable - application will treat it as Categorical.
There should be no more than 40 Categorical Variables.

Omit	Cat	Omit	Omit	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat	Cat
Errores Sig	Carrera	Padron	Sexo	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17
	9	84585	M	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	1	0
	10	84722	M	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0
	10	84653	M	0	1	1	1	0	0	0	0	0	0	1	1	0	0	0	1	0
	9	79600	M	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1
	10	84078	M	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	1	0
	10	84134	M	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	84870	M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	10	84620	M	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	85032	F	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
	10	85413	M	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	10	84962	F	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0
	9	84926	M	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	84734	M	1	1	1	1	0	1	1	0	0	0	0	1	0	0	0	0	0
	10	84091	M	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0
	10	82557	M	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0
	10	84690	M	0	0	1	1	0	0	0	0	1	1	0	1	0	0	0	0	1
	10	84087	M	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
	10	84587	M	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
	10	84371	M	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1

Figura 3: Pantalla ingreso de datos codificados.

6.1.1. El ingreso de los datos.

Se observa que existen tres tipos de datos que se ingresaron en forma codificada:

- *Categóricos*: son las palabras, que no son la salida esperada de la red.
- *Continuos*: son datos numéricos que representan números reales.
- De *Salida (Output):* son los valores que se desea que la red prediga, los que a su vez pueden ser categóricos ó continuos.

Cat	Cat	Cat	Cat	Cat	Cat			
E16	E17	E18	E19	E20	E21	Predicted Out	Score (aprobado)	Score (desaprobado)
0	1	0	0	10	0	aprobado	0,970997572	0,027049135
0	0	10	0	0	1	desaprobado	0,001156448	0,998752892
0	0	0	0	1	0	desaprobado	0,005972461	0,994004071
100	0	0	0	11	1	aprobado	0,979461491	0,019286392
0	0	10	0	101	101	aprobado	0,999360442	0,000711223
1	0	1	0	1	1	aprobado	0,944856524	0,058509205
1	0	0	0	1	0	aprobado	0,999345362	0,000726172
0	0	1	0	0	0	aprobado	0,544088364	0,455251902
1	11	1	0	1	0	aprobado	0,983933151	0,01474512
0	0	0	0	1	1	desaprobado	0,006141305	0,993789971
0	0	0	0	0	1	aprobado	0,680459857	0,331099778
0	0	0	0	0	0	aprobado	0,779129326	0,220500425
1	0	0	0	11	11	aprobado	0,999358594	0,000712385
1	0	0	0	1	1	aprobado	0,987725079	0,011409681
0	0	0	0	0	0	aprobado	0,994639397	0,005599276
0	0	0	0	1	1	aprobado	0,945939481	0,057353366
0	0	0	0	1	1	desaprobado	0,171479136	0,828849435
0	0	0	0	0	1	desaprobado	0,000927733	0,999024689
0	0	0	0	1	1	aprobado	0,954882205	0,047749929
0	0	0	0	10	1	aprobado	0,99622333	0,003323361
0	0	1	0	0	1	desaprobado	0,030887676	0,968400836

Figura 4: Pantalla predicción evaluación final

En los casos que se muestra en la Figuras 3 y 4, se muestra la salida que es de tipo categórica ya que será: *aprobado* ó *desaprobado*.

	Entrada	Acción	Salida
1	Datos de los errores de los alumnos de la materia en exámenes	Confección de base de datos de errores de cometidos	Errores codificados
2	Datos de entrenamiento	Entrenamiento de la red	Parámetros con error mínimo. Red entrenada
3	Datos de prueba	Aplicación de la red	Pronóstico
4	Errores codificados	Aplicación de la red	Diagnóstico

Tabla 1: Etapas y acciones seguidas

Predicted Output	Score (aprobado)	Score (desaprobado)
Aprobado	0,970997572	0,027049135
Desaprobado	0,001156448	0,998752892

Tabla 2: Ejemplo de Predicción Aprobado–Desaprobado

000	no se equivocó nunca
001	se equivocó en el parcial
010	se equivocó en el primer recuperatorio
100	se equivocó en el segundo recuperatorio

Tabla 3: Codificación de las Instancias de Aprobación

Por lo tanto, la red utilizará dos neuronas para ello; una cuya salida, a través de un número real, representa a los *aprobados* y otra, que también devuelve un número real, cuya salida está asociada a los *desaprobados.* En la primera fila del ejemplo de la Tabla 2 se observa que la para la salida de 0.970 la neurona está asociada a *"aprobado"* y sólo 0.027 la asociada a *"desaprobado",* por lo que la red da como resultado el primero de éstos. En la segunda fila se presenta el caso inverso. Los datos de las evaluaciones de los estudiantes (a través de 6 instancias de aprobación: un parcial con dos recuperatorios y tres oportunidades para el examen final) han sido codificados como se observa en la Tabla 3. El resto son combinaciones de las opciones presentadas. Con respecto a las columnas, E1 a E21 de la Figura 3, son los tipos de errores detectados en cada una de las evaluaciones agrupados luego del análisis realizado.

6.1.2. La selección de los parámetros de entrenamiento de la red.

En la Figura 5 se presentan los parámetros característicos de la red obtenidos. Se han explorado otras arquitecturas, con otras combinaciones de parámetros pero se obtuvieron errores mayores al 5%.

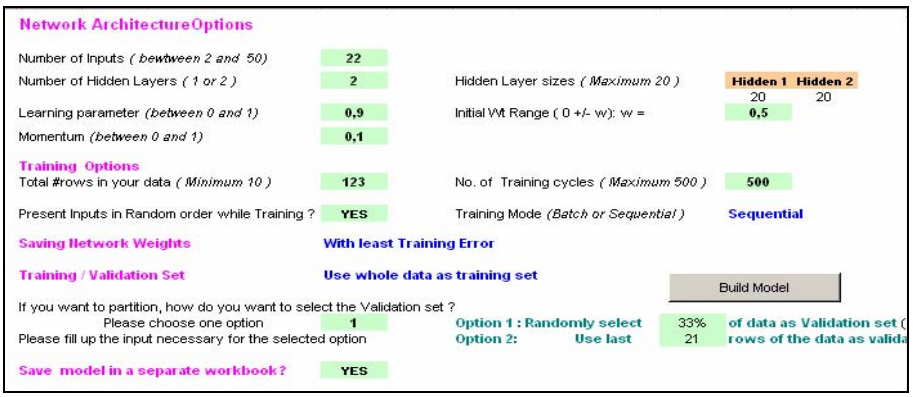


Figura 5: Características de la red

6.1.3. La predicción de la red

Cuando un estudiante se equivoca o no en alguno de los temas, aunque no haya completado aún todo el curso, es decir, cundo solo rindió la evaluación parcial y no accedió aún a la evaluación recuperatoria, por ejemplo, se puede predecir si aprobará o no en la instancia próxima. La idea es utilizar esta red como primer paso para el uso de una serie de dos redes. Cuando la primera red prediga que no aprobará, una segunda red del mismo tipo, basada en los errores cometidos, le puede indicar cuáles son los temas que debe estudiar, a fin de recomendarle los ejercicios por núcleos temáticos. Para este primer entrenamiento predictivo la red operó con un error del 4%, lo que es más que aceptable para el trabajo con grupos humanos [9].

6.1.4. Resultados de aplicación de la red

En la Figura 4 se observan las características de la red usada. Cuando el error es inaceptable, o sea mayor al 5%, se debe descartar la red y comenzar el proceso de entrenamiento nuevamente variando los distintos parámetros que la definen, entre los que se pueden citar el parámetro de aprendizaje α (el valor inicial y el régimen de modificación a lo largo de los ciclos, el momentum β, los pesos aleatorios iniciales, el vecindario gaussiano, la cantidad de atributos que se utilizarán para el entrenamiento, la cantidad de las observaciones que se utilizarán y las que se descartarán, justificando por qué se descartarán. El tiempo de entrenamiento de una red neuronal para los valores utilizados es de aproximadamente 1.5 horas en promedio. Se entrenaron redes con más de 600 variaciones en los parámetros hasta encontrar los más adecuados y los valores mínimos correspondientes al error.

7. UNA APLICACIÓN PARA ASISTIR A LOS ESTUDIANTES EN SUS AUTOEVALUACIONES

7.2.1. Descripción

A fin de ayudar en el proceso de reconceptualización de los errores, se elaboró una aplicación basada en la *web* que está disponible en los servidores de la universidad para que los estudiantes puedan efectuar sus autoevaluaciones. Esta herramienta cuyas pantallas se pueden ver en as Figuras 6 a 8), les permite evidenciar sus fallas a fin de que puedan tomar conciencia de sus limitaciones [34]. La autoevaluación está dirigida a estudiantes del área de Programación Básica. Como se cuenta desde el año 2000 con una base de datos de los errores más comunes de los estudiantes en sus evaluaciones, se pudieron construir series de preguntas con base en ellos donde cada grupo de preguntas se corresponde a cada Unidad Didáctica trabajada. Con esta instrumentación se buscó que el alumno pudiera darse cuenta de sus debilidades a fin de reforzarlas antes de su evaluación parcial o final. (En el ANEXO se presentan los diferentes tipos de preguntas, con sus respuestas y con las observaciones de cada caso).

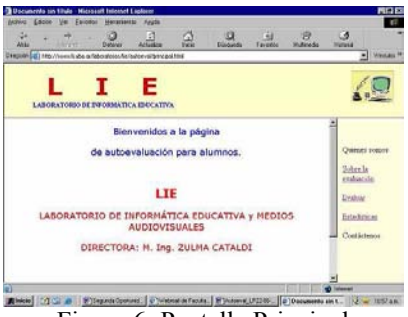


Figura 6: Pantalla Principal

Figura 7: Los contenidos.

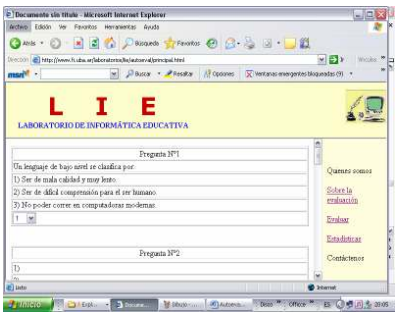


Figura 8: La autoevaluación

La aplicación *Autoeval*, es un programa muy simple que permite a los alumnos autoevaluar sus conocimientos mientras los profesores pueden ver los datos estadísticos en cualquier momento para tener un panorama general respecto del nivel del curso que está interactuando. El software está basado en formularios HTML, *scripts* PHP y una pequeña base de datos MySQL, utilizando de esta manera herramientas de muy bajo costo y sin problemas legales. Básicamente, se muestra a los alumnos un formulario HTML con *n* preguntas de tipo *múltiple choice*, dándoles 3 opciones a elegir, más la opción de no contestar. Las preguntas han sido formuladas considerando los datos estadísticos acerca de los errores más frecuentes de los alumnos en las evaluaciones parciales y finales. Al finalizar su autoevaluación, el alumno realiza un *"submit"*, llamando así a un *script* que corrige el examen mostrándole al estudiante la cantidad de respuestas correctas, y las opciones correctas de los puntos que haya contestado erróneamente. El programa[3] consiste de las siguientes

[3] *El modelo de datos:* En las cátedras donde se utilizará el programa se han desarrollado los contenidos en 10 Unidades Didácticas (o Capítulos), por lo cual se podría haber optado por usar 10 tablas en la base de datos, donde cada una almacenara los ítems del capítulo que le correspondiera, así, sería un modelo muy intuitivo y ordenado, sin embargo, para independizar el programa de la estructura de datos se optó por tener una única tabla de preguntas con la estructura de la Tabla 1, donde sabiendo que $chapter contiene el número de capítulo deseado, en #1 se genera la consulta que se

etapas: a) creación de registros (carga de las preguntas de examen), b) presentación de examen, c) corrección y d) presentación de estadísticas [34].

NUM	Int unsigned auto_increment **Primary key**
Question	Text
Ans1	Text
Ans2	Text
Ans3	Text
Chapter	Char(5)
correct	Char(1)

Tabla 1: Estructura de la base de Datos[4]

```
#1 $query = "select * from  questions where
chapter like '%".$chapter."%' ";
#2 $result= mysql_query($query);
#3 $num_rows= mysql_num_rows($result);
#4$top= rand(1,$num_rows);
#5 for ($i=0; $i< $top ; $i++)
            {
            $row= mysql_fetch_array($result); }
#6 return $row;
```

Tabla 2: Selección de las preguntas

De esta manera el programa seleccionará para cada Unidad Didáctica una pregunta como se ve en la Tabla 2. En cuanto a las *estadísticas* el programa muestra la cantidad de exámenes corregidos, la cantidad de exámenes con más de 6 respuestas correctas, la cantidad de respuestas incorrectas por cada Unidad Didáctica y su respuesta correcta.

7.2.2. Resultados

La aplicación se probó con alumnos voluntarios desde 2005. A partir de esta primera exploración se ha visto que los alumnos fueron capaces de elaborar preguntas e inquietudes con mayor frecuencia y relevancia. Por otra parte, se convirtieron en agentes de apoyo a sus compañeros ya que pudieron ayudarlos en los inconvenientes de menor envergadura que ellos fueron capaces de reelaborar merced a su autoevaluación. Los docentes lograron modelizar algunos esquemas de los errores que se presentaron con mayor frecuencia. A partir de ahí, se observó que los alumnos trabajaron de manera más autónoma, la temática de las preguntas cambió desplazándose de los temas instrumentales como el uso del lenguaje hacia temas que tenían que ver con la manera de optimizar los algoritmos que ellos creaban. Por otra parte, el índice de aprobación ha mejorado sensiblemente.

8. CONCLUSIONES Y TRABAJOS FUTUROS

Se trabajó en el sistema de predicción donde a partir del rendimiento obtenido en las evaluaciones parciales se puede predecir cómo será la *"performance"* en futuras evaluaciones. Los datos obtenidos permitirán diseñar un sistema recomendador de modo que el alumno ante la situación de ser examinado pueda acceder al sistema quien le sugerirá una serie de ejercicios y problemas para que pueda internalizar los errores que fueron clave para la reprobación de las evaluaciones previas. Se busca brindar a los estudiantes una herramienta para que puedan tomar conciencia de sus propios errores y no cometer las mismas fallas en las evaluaciones finales. Este objetivo está cumplido, pero se ha visto que el impacto podría ser de mayor envergadura con el tiempo si se consideran los resultados que se presentaron. Quedó establecido que, indudablemente, la *autoevaluación* resulta un acercamiento, entre aquellos que son posibles, hacia la autonomía del alumno y la mejora del proceso de aprendizaje.

Como líneas futuras de investigación se prevé trabajar en el diseño del sistema recomendador de ejercicios y problemas de modo que el estudiante pueda adquirir una cierta autonomía en la preparación de sus exámenes finales. Como trabajos posteriores se propone: a) Ampliar los contenidos disponibles para la autoevaluación, b) Escalar el sistema informático para que se pueda realizar un seguimiento del alumno, de esta forma el docente puede tener una *clusterización* de las necesidades cognitivas de su clase. c) Escalar el sistema informático de tal forma que evolucione hacia bases de datos e interfaces capaces de interactuar con el alumno de manera autónoma y d)

quiere realizar en un string, en #2 se ejecuta la consulta, y se devuelve en $result todos los registros donde el campo capítulo sea el buscado ($chapter). En #3 se calcula la cantidad de registros obtenidos en #2 ($num_rows), para luego generar un número aleatorio entre 1 y $num_rows y en #5 acceder al registro que posee ese número y finalmente devolverlo al programa principal en #6.

[4] *"num"* es la clave única identificatoria del registro, *"question"* es la pregunta en sí, *"ans1" "ans2" y "ans3"* son las respuestas posibles a *"question", "chapter"* es el número de capítulo (unidad didáctica) en cuestión y finalmente *"correct"* es el número de respuesta correcta (1, 2 ó 3).

Integrar los hallazgos en el módulo de evaluación para un Sistema Tutor Inteligente cuya arquitectura se está desarrollando.

9. AGRADECIMIENTOS

Esta comunicación forma parte de los proyectos de investigación: *Sistemas inteligentes aplicados a la predicción del comportamiento de los estudiantes y diagnóstico* 2005-2006 LIE-DC/04-07 del Laboratorio de Informática Educativa y Medios Audiovisuales (LIEMA) de la Facultad de Ingeniería, de la Universidad de Buenos Aires y 25/C099 *Modelado del tutor basado en redes neuronales para un Sistema Tutor Inteligente,* de la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional 2007-2008 convenio FI-UBA y UTN-FRBA. Los autores agradecen a los alumnos que participaron de la experiencia.

10. REFERENCIAS

[1].Lage, F.; Cataldi, Z. y Denazis, J. M. (2000). *The Scripts of University Students and Experts in the Preparation of the Examinations: A Study in Process*. FIE 2000: 30th ASEE/IEEE Frontiers in Education Conference, Kansas City Missouri, 18-21 de octubre. Paper 1154. Proceedings en CD-ROM. ISBN 0-7803-6242/0/

[2].Copello, G.; Cataldi, Z. y Lage, F. (1999). *La comprensión de los errores.* Proceedings del V Congreso Internacional de Ingeniería Informática. Páginas 210-217. Editado por Departamento de Publicaciones de la Facultad de Ingeniería.

[3].Pozo, J. I. (1998). *Teorías cognitivas del aprendizaje*. Morata.

[4].Pozo, J. I. (1999). *Aprendices y Maestros*. Alianza.

[5].Ausubel, D.; Novak, J. y Hanessian, H. (1983) *Psicología educativa: un punto de vista cognitivo.* 2ª Ed. México: Trillas. 624p.

[6].Ohlsson, S. (1996) *Learning from performance of errors.* Psychological Review 3 (2) p. 241-262.

[7].Khuwaja, R.A. (1994) A Model of Tutoring: Facilitating Knowledge Integration Using Multiple Models of the Domain. *Ph.D., Illinois Institute of Technology*

[8].Giraffa, L.M.M.; Nunes, M. A.; Viccari, R.M. (1997) *Multi-Ecological: an Learning Environment using Multi-Agent architecture*. MASTA'97: Multi-Agent System: Theory and Applications. Proceedings. Coimbra: DE-Universidade de Coimbra.

[9].Cataldi, Z. 2005. *Sistemas tutores inteligentes: los estilos del estudiante para selección del tutorizado.* WICC 2005. 13 y 14 de mayo. Universidad Nacional de Río Cuarto. Córdoba. RED UNCI

[10].Salgueiro, F. A, Costa, G., Cataldi, Z., García Martinez, R. y Lage, F. J. 2005. *Sistemas inteligentes para el modelado del tutor.* GCETE'2005**,** Global Congress on Eng. and Technology Education. marzo 13-15

[11].Abbas, H. (1998) Designing a New Domain Knowledge Base for an Intelligent Tutoring System. *Ph.D., Illinois Institute of Technology.*

[12].Brachman, R.J. (1988) *The basis of knowledge representation and reasoning.* AT&T. Technical Jurnal. 67, 1:15.

[13].Brachman; R.J. (1985) On the epistemological status of semantic networks. En Brachman, R. y Levesque, H. (Eds.) *Readings in knowledge representation* (191-215). Los Altos. Morgan Kaufman Pub. Inc.

[14].Viccari, R. M. (1993). Inteligência Artificial e Educação: Indagações Básicas. IV Simpósio Brasileiro de Informática e Educação.

[15].Viccari, R.M. y Giraffa, L.M. (1996). Sistemas Tutores Inteligentes: Abordagem Tradicional x Abordagem de Agentes. XIII Simpósio Brasileiro de Inteligência Artificial, Curitiba.

[16].Casas, M. (1999) *contribuições para a modelagem de um ambiente inteligente de educação baseado em realidade virtual.* Tesis Doctoral Univ. de Federal de Sta. Catarina. Pós-graduação em Eng. de Produção.

[17].del Brío, B. M. y Sanz Molina, A. (2001) *Redes neuronales y sistemas difusos.* Paraninfo.

[18].Kohonen, T. (1988) Self-Organizing Maps Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, NY(pp 236)

[19].Hilera González; R. y Martínez Hernando, A. (2000) Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones. Ra-ma, Madrid.

[20].Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. (1986). *Learning internal representations by back-propagating errors in Parallel Distributed Processing*: *Explorations in the Microstructure of Cognition*. Eds. Cambridge, MA: MIT Press, vol. 1, p. 318-362.

[21].Rosenblatt, F. (1958), *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review, 65, 386-408.
[22].Palmer, A., Montaño, J.J. y Jiménez, R. (2001) *Tutorial sobre Redes Neuronales Artificiales: El Perceptrón Multicapa.* Revista Electrónica de Psicología Vol. 5, No. 2, Julio ISSN 1137-8492.
[23].Smith, M. (1993). *Neural networks for statistical modeling*. New York: Van Nostrand Reinhold.
[24].Masters, T. (1993). *Practical neural networks recipes in C++*. London: Academic Press
[25].Saha, A. (1998) *Application of Ridge Regression for Improved Estimation of Parameters in Compartmental Models;* Tesis Doctoral. Departamento de Estadística.
[26].Ausubel D., Novak J. y Hanesian H.(1993, 1997). *Psicología educativa. Un punto de vista cognitivo.* Trillas. Décima impresión
[27].Hernández Rojas, G. (1998) *Paradigmas en Psicología de la educación*. Paidós. México
[28].House, E. R. (1997) *Evaluación, ética y poder,* Ed. Morata: Madrid
[29].Angulo, J. F. y Blanco, L. (coordinadores) (1994). *Teoría y desarrollo del currículum"*. Ediciones Aljibe. Málaga.
[30].Blanco Prieto, F. (1994). *La evaluación en la Educación Secundaria.* Amarú. Salamanca.
[31].Gimeno Sacristán, J. (1981). *Teoría de la enseñanza y desarrollo del currículo.* Anaya. Madrid.
[32].Zimmerman, B. J. (1989) A social cognitive view of self-regulated academic learning. *Journal of Educational Psychology,* 81(*3*), 329-339.
[33].Zimmerman, B. J. (1994) Dimensions of academic self-regulation: a conceptual framework for education. En D. H. Schunk y B. J. Zimmerman (Eds.), *Self-regulation of learning and performance. Issues and educational applications.* Hillsdale, NJ: Erlbaum.
[34]. Cataldi, Z.; Méndez, P. y Lage, F: 2006. *La autoevaluación en Programación Básica en carreras no informáticas desde la perspectiva constructivista del aprendizaje autorregulado.* JENUI 2006. XII Jornadas de Enseñanza Universitaria de la Informática. Bilbao, 12 a 14 de julio. Thomson. 279-286.

ANEXO: Algunas preguntas del programa

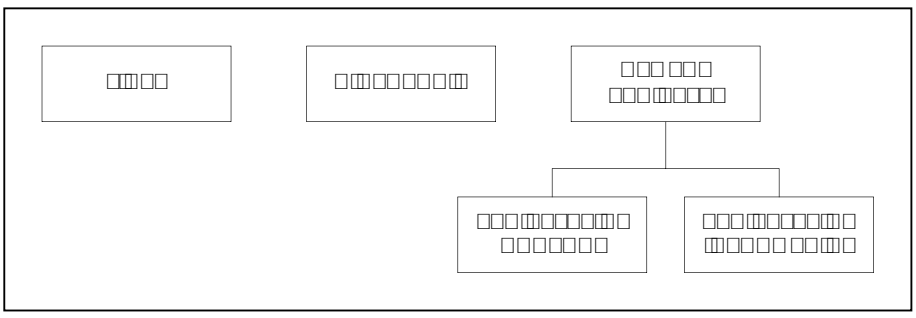
Unidad Didáctica 3: *Estructuras secuenciales.* Indique cuál es la respuesta correcta
Se tiene el precio y nombre de un producto en una variable de tipo real (precio) y en una variable de tipo string (s_nombre) respectivamente. Se desea mostrar en pantalla el nombre y el valor de dicho producto. Seleccione la manera óptima de hacerlo.

a) Writeln('El precio de ', s_nombre,' es de $',precio);
b) Write('El precio de ');
Write(s_nombre);
Write(' es de ');
Writeln(s_precio);
c) Writeln('El precio de s_nombre es de $precio');

La respuesta correcta es a.

Unidad Didáctica 6: *Subprogramas*. Indique cuál es la respuesta correcta. Se tiene el siguiente procedimiento
procedure sumados(a: integer);
begin
a:= a+2;
end;
Si se lo llama desde el programa principal siendo en él a=8, cuál será el valor de a luego de ejecutarse el procedure?
a) 8 b) 10 c) 0
El resultado correcto es a.
Nota: El estudiante debe darse cuenta que el parámetro es pasado por valor y no por referencia.

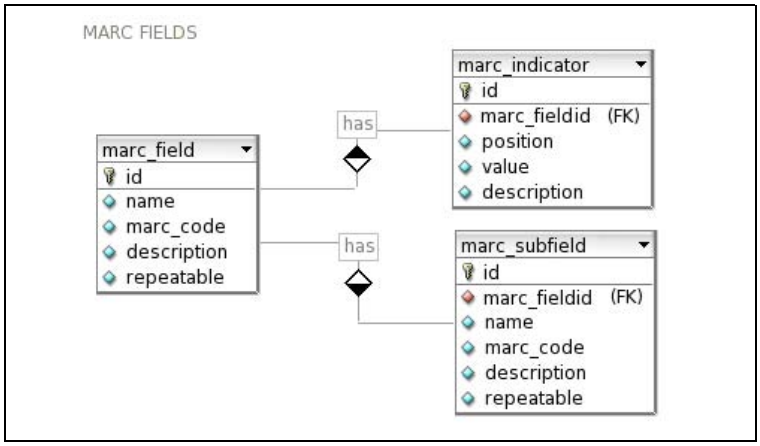
-
-
-
-
-
-
-
-



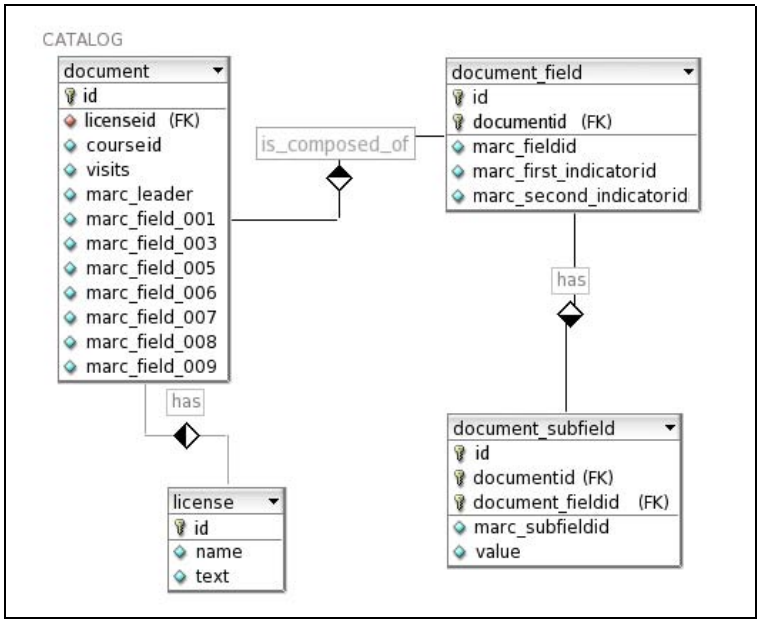
-
-
-
-
-
-
-

-
-
-

-
-
-
-
-


MARC FIELDS
marc_field
id
name
marc_code
description
repeatable
has
has
marc_indicator
id
marc_fieldid (FK)
position
value
description
marc_subfield
id
marc_fieldid (FK)
name
marc_code
description
repeatable

-
-
-
-
-
-
-
-


CATALOG
document
id
licenseid (FK)
courseid
visits
marc_leader
marc_field_001
marc_field_003
marc_field_005
marc_field_006
marc_field_007
marc_field_008
marc_field_009
is_composed_of
document_field
id
documentid (FK)
marc_fieldid
marc_first_indicatorid
marc_second_indicatorid
has
has
document_subfield
id
documentid (FK)
document_fieldid (FK)
marc_subfieldid
value
license
id
name
text

Hacia un modelo de evaluación de entornos virtuales de enseñanza y aprendizaje. La importancia de la usabilidad.

Ariel Ferreira Szpiniak[1]
aferreira@exa.unrc.edu.ar
Cecilia Sanz[2]
csanz@lidi.info.unlp.edu.ar

Abstract
This paper presents some research aspects which are being carried out as part of a thesis of the Master in Computer Technology applied to Education, at the School of Computer Sciences of the UNLP.
In particular, it is of interest to study and analyze the current Teaching and Learning Virtual Environments (TLVE), how they have evolved and which the criteria allowing the evaluation of such environments are.
This paper describes different evaluation models of TLVE, presented by different authors. They are compared and summed up to give a context to this work. Then, the concept of usability and usability heuristics is reviewed. Finally, some ideas to reach an evaluation model of TLVE are proposed, incorporating usability criteria. Even though there does not yet exist applications of the proposed evaluation model, we intend to apply it for a TLVE in particular, in a specific institutional context.
Keywords: Teaching and Learning Virtual Environments, Usability, TLVEs Evaluation.

Resumen
En este trabajo presentamos algunos aspectos de la investigación que se está realizando como parte de un trabajo de tesis de la Maestría de Tecnología Informática Aplicada en Educación, de la Facultad de Informática de la UNLP.
En particular, es de interés estudiar y analizar los actuales Entornos Virtuales de Enseñanza y de Aprendizaje (EVEA), cómo han evolucionado y cuáles son los criterios que permiten evaluar dichos entornos.
Se presentan aquí diferentes modelos de evaluación de EVEAs, presentados por diversos autores. Se los compara y resume para dar contexto a este trabajo. Se revisa, luego, el concepto de usabilidad y de heurísticas de usabilidad. Se proponen, finalmente, algunas ideas para alcanzar un modelo de evaluación de EVEAs, que incorpore criterios de usabilidad. Si bien no existen aún aplicaciones del modelo de evaluación propuesto, se tiene la intención de aplicarlo para un EVEA en particular, en un contexto institucional específico.
Palabras clave: entornos virtuales de enseñanza y de aprendizaje, usabilidad, evaluación de EVEAs.

1. Entornos virtuales enseñanza y aprendizaje

Los entornos virtuales de enseñanza y aprendizaje (EVEA) son aplicaciones informáticas desarrolladas con fines pedagógicos, es decir, persiguen un fin educativo. Fueron diseñados con el propósito de facilitar la comunicación pedagógica entre los participantes en un proceso educativo, fundamentalmente docentes y alumnos, sea éste completamente a distancia, presencial, o de naturaleza mixta, es decir, que

[1] Departamento de Computación. Universidad Nacional de Río Cuarto, Ruta 36 Km 601, 5800, Río Cuarto, Córdoba, Argentina

[2] III LIDI, Facultad de Informática - Universidad Nacional de La Plata. Calle 50 y 115, 1900, La Plata, Buenos Aires, Argentina

combine ambas modalidades en diversas proporciones (blended). Funcionan utilizando redes telemáticas como soporte, principalmente Internet.
Los EVEA ponen el énfasis en la interactividad como estrategia para favorecer los procesos educativos. Deben ser lo suficientemente versátiles como para no condicionar la propuesta pedagógica y permitir un amplio abanico de posibilidades en cuanto a los modelos susceptibles de ser utilizados, desde un modelo centrado en el docente hasta un modelo centrado en el alumno [2]. Es decir, lo importante es que el diseño tecnológico acompañe al modelo pedagógico, sin perder de vista que la herramienta tecnológica solamente, aunque sea la mejor, no garantiza el cumplimiento de los procesos educativos [3].
Para conseguir estos objetivos, los EVEA proveen herramientas que se ocupan de la gestión de usuarios, gestión de cursos y gestión de servicios de comunicación y distribución y tratamiento de contenidos.
Los EVEA surgieron con la masificación del uso de Internet y las potencialidades de la misma como vehículo para el aprendizaje. En sus comienzos el uso de Internet se limitaba a la explotación independiente de sus servicios (www, ftp, e-mail, foros, chat, etc.), donde la modalidad predominante consistía en publicar o enviar a los alumnos los contenidos del curso, generalmente divididos en módulos, y luego resolver las dudas o discutir los contenidos a través del contacto por correo electrónico entre el alumno y el docente y/o una lista de distribución [4].
A los efectos de proveer un soporte más eficaz para diferentes modalidades educativas, con el paso del tiempo la tecnología informática generó una serie de entornos, ambientes o herramientas genéricas que sirvieron de base, para optimizar las distintas fases del proceso de enseñanza y aprendizaje. La principal ventaja ofrecida por este tipo de entornos ha sido la integración de diferentes herramientas y servicios para gestionar y desarrollar actividades de formación. En los últimos años, la integración y centralización en un único entorno de estas herramientas, ha facilitado y optimizado mucho la labor tanto de administradores como de docentes, generando además entornos favorables para el trabajo de los alumnos.
Los EVEA sirven para distribuir materiales educativos en formato digital (textos, imágenes, audio, simulaciones, juegos, etc.) y acceder a ellos, para realizar debates y discusiones en línea sobre aspectos del programa de la asignatura, para integrar contenidos relevantes de la red o para posibilitar la participación de expertos o profesionales externos en los debates o charlas. Estos sistemas son un soporte tecnológico donde se integran y combinan diversos servicios, herramientas y recursos de Internet para favorecer distintas fases del proceso de enseñanza y de aprendizaje tales como planificación, implementación, desarrollo y evaluación del currículum [5].

Existe una gran cantidad de entornos virtuales orientados a la educación superior. En algunos casos se habla de más de ciento treinta [7], en otros, si se suman los desarrollos propios de las universidades, el número aumenta a varias centenas [8]. Muchos de ellos, se utilizan en ámbitos específicos o muy reducidos, otros se han fusionado o discontinuado en su desarrollo. Algunos, al no tener un fin comercial, no trascienden y es muy difícil conocerlos en detalle. Luego de la “fiebre de entornos” que se produjo hacia fines del siglo XX, donde cada institución o empresa creaba su propia solución, sobrevino otra etapa caracterizada por un análisis pormenorizado de los Entornos Virtuales de Enseñanza y Aprendizaje desde la óptica educativa. Investigadores en educación, pedagogos y docentes criticaron el excesivo peso que hasta el momento se había puesto sobre los criterios técnicos en desmedro de los pedagógicos, didácticos y comunicacionales. Con el paso del tiempo se produjo por un lado, una decantación, y por el otro una vertiginosa evolución de los entornos que contaban con mayor calidad y apoyo, tanto desde el punto de vista institucional, económico, de la comunidad de usuarios o desarrolladores. En la actualidad, dentro del ámbito internacional encontramos aproximadamente 40 productos de este tipo que han logrado preponderancia.

Los entornos se utilizaron exclusivamente en un primer momento para la formación a distancia, pero en los últimos años se han incorporado muy fuertemente a la educación presencial. Por medio de estos entornos se han logrado aprovechar las características de accesibilidad y cooperación entre los usuarios de la red, orientando fundamentalmente el proceso educativo hacia nuevas manera de aprender, apoyadas en el trabajo colaborativo, acompañadas de una serie de recursos, más allá de la mera presencia del docente [9].

Así, la primera generación de entornos basados en la distribución de materiales y la evaluación mediante pruebas automáticas, ha dado paso a una segunda generación de entornos inspirados en los nuevos conocimientos sobre cómo se aprende usando recursos en línea. Esta segunda generación está más orientada hacia la comunicación didáctica, el diseño y monitorización del flujo de trabajo colectivo con los materiales y, en general, hacia la actividad didáctica o experiencia de aprendizaje. Todo ello sin olvidar la necesaria estandarización de los contenidos, si pretendemos reutilizarlos y compartirlos [5].

Cada entorno está diseñado, consciente o inconscientemente, desde una filosofía pedagógica. Algunos entornos privilegian la creación y distribución de contenidos formativos, asumiendo que el contacto de los estudiantes con materiales relevantes, cuidadosamente diseñados, es el elemento clave del aprendizaje. Otros, en cambio, potencian la comunicación entre los participantes en la convicción de que el aprendizaje es producto de la interacción social y la construcción compartida de significados en un ambiente rico en información y en oportunidades de conocimiento. Actualmente es posible observar tres tendencias bien [5]:

- Entornos centrados en la creación, gestión y distribución de contenidos que además incorporan algunas herramientas de comunicación.
- Entornos centrados en la comunicación y las actividades de enseñanza / aprendizaje que incluyen además herramientas para gestionar materiales.
- Entornos de trabajo en grupo para comunidades académicas que agregan algunas funcionalidades utilizables en la enseñanza.

Por ello, los distintos módulos, componentes, recursos o herramientas incorporadas en los entornos poseen distintas funcionalidades y están dispuestos de modo diferente. Sin embargo, un buen entorno debe ser lo sumamente flexible como para no imponer un estilo docente en particular sino posibilitar la mayoría de ellos a fin de que los docentes y los estudiantes lo experimenten como un aumento de posibilidades y no como una limitación, un elemento distorsión o un freno a su creatividad y a su capacidad de innovación [5].

2. ¿Cómo evaluamos los EVEAs?

En esta sección presentaremos una breve descripción en cuanto a lo referido por diferentes autores sobre cómo podemos evaluar los EVEAs. Esto permitirá contextualizar la propuesta realizada en este trabajo.

En la actualidad los EVEA poseen herramientas suficientes como para desarrollar acciones formativas de diverso tipo. No obstante se observan ciertos inconvenientes y limitaciones que dificultan su implementación [5]. A los efectos de determinar objetivamente las potencialidades de cada uno, es necesario realizar un proceso de evaluación de los mismos. La evaluación está orientada a valorar la calidad del entorno virtual a través del cual se implementan acciones formativas [10]. Mediante la evaluación podremos determinar su filosofía pedagógica, su flexibilidad didáctica, sus funcionalidades y usabilidad.

Los principales indicadores que pueden tenerse en cuenta a la hora analizar un entorno virtual están relacionados con el tipo de actividad educativa a desarrollar (cursos, carreras, comunidades virtuales), la modalidad, las características de los destinatarios y de las posibilidades y limitaciones técnicas disponibles (ancho de banda, equipamiento informático y disponibilidad de tiempo, entre otros). Existen gran cantidad de enfoques e instrumentos para llevar a cabo esta tarea. La mayoría están orientados a

determinar las características de los entornos en función de una serie de categorías de análisis como características generales, requerimientos técnicos, herramientas para la enseñanza y el aprendizaje, herramientas para la gestión y administración, etc. Aunque las categorías varíen de un enfoque a otro, es muy común encontrar cierta recurrencia en las cuestiones centrales. Algunos instrumentos de evaluación son netamente técnicos, a partir de una lista de chequeo de funcionalidades, servicios y herramientas, y otros adoptan miradas desde lo educativo a partir de las potencialidades que poseen a efectos pedagógicos y didácticos. En su mayoría se trata de instrumentos para evaluar y comparar entornos a los efectos de seleccionar el más adecuado para ser utilizado en un contexto particular. Podríamos decir que encontramos en un extremo a los planteos de índole exclusivamente tecnológica y generalmente cuantitativos, en el otro extremo a los planteos educativos y generalmente cualitativos. Entre ambos existe gran variedad de matices. Algunos de los modelos de evaluación más importantes serán sintetizados a continuación.

2.1 Revisión de algunos modelos

El modelo desarrollado por Adell [5] sostiene que los criterios planteados están pensados conforme con las necesidades y los tipos de uso educativo que se pretende realizar del EVEA. Se trata de un análisis de corte cualitativo realizado en función de tres criterios: flexibilidad didáctica, usabilidad y flexibilidad tecnológica. La flexibilidad didáctica se enfoca hacia las facilidades para el diseño del currículo (acceso a los cursos, organización del curso de manera temporal, por temas, etc.) y las herramientas de aprendizaje disponibles (foros, materiales, mensajería, chat, ejercicios, deberes, extensibilidad, multimedia, grupos, copias de seguridad, compatibilidad de contenidos, seguimiento y evaluación del aprendizaje). La usabilidad tiene en cuenta a la facilidad de uso, los conocimientos técnicos de los usuarios, ayuda, trabajo offline, accesibilidad e idiomas. Por último, la flexibilidad tecnológica tiene en cuenta los siguientes aspectos: requisitos técnicos, escalabilidad, posibilidades de integración con otros sistemas, mecanismos de autenticación de los usuarios, adaptación visual y costes de implementación, administración y mantenimiento.

Por su parte, Pallof y Pratt [11] sugieren tomar en cuenta cinco aspectos que deberían estar presentes en un buen entorno: interacción, introspección, innovación, integración e información. La **interacción** tiene que ver con los aditamentos necesarios para establecer procesos comunicacionales sincronizo y asíncronos entre todos los participantes: correo electrónico, enlaces web, foros de discusión, chats, desarrollo de páginas web personales, etc. La **introspección** se asocia con la posibilidad de que las herramientas comunicacionales permitan generar el desarrollo de pensamiento crítico y creativo por medio de una serie de recursos, en donde se acompañe de audio, video y texto, junto con espacios para actividades a desarrollar por parte de los estudiantes. Sin embargo, la propuesta didáctica sigue siendo un elemento fundamental en estos casos, ya que la herramienta puede contar con las mejores características pero ser usada de forma inadecuada. La **innovación** permite al profesor generar actividades diferentes a las acostumbradas en una clase convencional, permitiendo a los estudiantes participan cada vez más, sacando provecho de su propio estilo de aprendizaje. De la misma manera, el proceso de evaluación del curso amplía su gama de alternativas, gracias al apoyo de la tecnología. La **integración** tiene que ver con la manera en que los recursos tecnológicos incorporan la información a un proceso de generación de conocimientos, partiendo de una base relacional (entre todos los participantes del curso), lejos de posturas eminentemente individualistas. La **información** refiere a la posibilidad de contar con aquellos hechos o datos necesarios para el buen desarrollo del curso a la mano del estudiante, sin importar en qué lugar se encuentre: bases de datos especializadas, bibliotecas digitales, portafolios electrónicos, datos del curso, producciones del docente, etc.

También el Gabinete de Tele-Educación de la Universidad Politécnica de Madrid [12] ha elaborado una serie de indicadores para evaluar entornos para la formación por Internet. Estos indicadores, agrupados en cinco niveles, pretenden ser una ayuda para establecer una relación entre los objetivos que se buscan

y la oferta disponible. Ellos son: información técnica, edición de materiales, proceso de enseñanza – aprendizaje, administración y gestión académica, y, por último, otras características. Dentro de información técnica se consideran a su vez las características generales, la compatibilidad y la robustez. Las características generales son una serie de indicadores que posibilitan determinar la adaptabilidad a las necesidades y posibilidades institucionales tales como idiomas, número de usuarios permitidos, número de cursos, arquitectura escalable, precio y servicio técnico. La compatibilidad evalúa el carácter universal del entorno mediante parámetros como: requisitos de hardware y software, soporte de formatos multimedia, interoperabilidad con una infraestructura existente, estándares y accesibilidad. La robustez se ocupa de analizar las fallas derivadas de un mal diseño del software, la tolerancia ante utilización poco experta y la capacidad de controlar los accesos a las bases de datos y herramientas que integran el sistema, así como también la propiedad y protección de los materiales curriculares. El nivel de edición de materiales evalúa la edición de contenidos, la edición de elementos de evaluación y la posibilidad de creación de otros elementos complementarios como actividades de aprendizaje, bibliografía, enlace, glosario y índice de los contenidos del curso. Proceso de enseñanza – aprendizaje se refiere a los indicadores necesarios para evaluar las prestaciones que ofrece la plataforma en lo referente a: desarrollo y funcionamiento de los cursos, herramientas que dispone el profesor para efectuar una monitorización del trabajo de los alumnos, comunicación entre los alumnos y entre éstos y los profesores y herramientas que dispone el alumno para la realización de sus actividades de aprendizaje. El nivel de administración y gestión académica evalúa las herramientas para la gestión de usuarios, organización académica e informes. Por último, dentro de otras características, se encuentran a una serie de indicadores que pretenden evaluar aspectos subjetivos y que están relacionados con la confianza que ofrece el sistema y con su calidad ergonómica. Es por ello se evalúan aspectos tales como la evolución del entorno, quienes son sus propietarios, la opinión y aceptación de terceras personas, la sencillez de manejo, la comodidad y amabilidad de uso, la existencia de ayudas, documentación, recursos y capacitación sobre el uso del sistema. Como puede observarse, en este caso existe una mayor preponderancia de análisis de elementos más referidos a lo tecnológico y algunas cuestiones relacionadas con la usabilidad.

Basándose en investigaciones propias y de otros autores, Torres Toro y Ortega Carrillo [4] elaboraron una serie de indicadores de calidad a tener presentantes en los entornos de formación virtual. Ellos proponen cuatro categorías para analizar la calidad: técnica, organizativa y creativa, comunicacional y didáctica. En la categoría de calidad técnica se tienen en cuenta los aspectos para garantizar la solidez y estabilidad de los procesos de gestión y de enseñanza-aprendizaje. Entre los más importantes se destacan la infraestructura tecnológica necesaria, accesibilidad y complejidad, número de usuarios, coste de mantenimiento, conocimientos necesarios para su utilización, interfase sencilla, seguridad y acceso a los materiales, eficacia de gestión de cursos, licencias de uso o gratuidad. En cuanto a la calidad organizativa y creativa se detallan aspectos como la flexibilidad en la elección del proceso de enseñanza-aprendizaje, la posibilidad de adaptación a otros entornos formativos como la educación no formal o comunidades virtuales de aprendizaje, posibilidad de organizar automáticamente los contenidos en índices y mapas conceptuales, glosarios, integración de elementos multimedia, herramientas de evaluación, búsqueda de contenidos, seguimiento del curso, mantenimiento y actualización. En la categoría calidad comunicacional se analizan las posibilidades de comunicación asincrónica (foros, correo, noticias y calendario) y sincrónica (chats, audio y video conferencia) entre docente-alumno, alumno-alumno y docente-docente. Dentro de calidad didáctica, tiene en cuenta las principales aportaciones de las teorías cognitivistas y constructivistas al diseño de materiales curriculares virtuales como el principio del aprendizaje activo, principio del aprendizaje inductivo por descubrimiento, principio de la significatividad del aprendizaje, principio del aprendizaje cooperativo, principio de la versatilidad de los ambientes formativos, principio de la autonomía organizativa y del equilibrio cognoscitivo, principio de la secuencialidad conceptual, principio del andamiaje

cognoscitivo, principio del orden y la claridad didáctica y el principio de la comunicación multimedia eficaz.

El modelo de evaluación propuesto por Zapata Ros [13], se apoya en cinco categorías que posibilitan examinar de forma muy pormenorizada las características de un entorno. En general, cada categoría se desglosa en una serie de preguntas o ítems que pueden ser identificados o no dentro de la herramienta analizada. Ellos son características básicas, utilidades que generan ambientes de comunicación y de trabajo, funciones formativas que permite desarrollar, roles que se identifican y evaluación sobre la intervención psicopedagógica del sistema que soporta.

Pere Marquès elaboró una serie de criterios de calidad para los sistemas de teleformación [14] donde no se abordan únicamente las herramientas sino también la calidad de los materiales formativos, la labor de los docentes y tutores, el plan docente y la metodología del curso. Esta concepción sostiene que la calidad del entorno dependerá en gran manera de la actuación de los docentes que interactúan con los estudiantes en el marco de un determinado modelo pedagógico. Los criterios están agrupados en: aspectos relacionados con el entorno telemático, aspectos funcionales del curso on-line, aspectos relacionados con el plan docente y el modelo pedagógico, aspectos relacionados con los servicios complementarios.
El estudio realizado por Whitmyer Grimes [15], denominado Comparative Features Analysis of Leading Course Management Software, identifica un extenso conjunto de características deseables, 94 en total, para dar un correcto soporte a la enseñanza y el aprendizaje en línea. El propósito de sus investigaciones está orientado en dos sentidos: por un lado ayudar a las instituciones académicas a tomar decisiones sobre el entorno más conveniente a utilizar, y por el otro para orientar a los desarrolladores de este tipo de herramientas respecto a nuevas funcionalidades que sería provechoso fueran incorporadas en el futuro no muy lejano.

En general, la mayoría de los modelos presentados están orientados hacia las funcionalidades que ofrece el entorno. Se trata de modelos donde el usuario final no participa sino que son expertos quienes realizan la evaluación, asumiendo los diferentes roles para los cuales el entorno será utilizado. Esto se debe a que dichos instrumentos fueron pensados principalmente para la toma de decisiones sobre el entorno a adoptar, por parte de las universidades o de los centros de formación. Por ello, en la mayoría de los casos, se realiza una extensa categorización de características deseables o ideales que debería tener un sistema de este tipo. En la actualidad, la mayoría de los entornos que han logrado buenos niveles de aceptación disponen de las características buscadas en estos modelos de evaluación.

3. La importancia de considerar criterios de “usabilidad”

Como hemos mencionado anteriormente, los EVEAs son aplicaciones informáticas del tipo cliente/servidor que funcionan vía red, generalmente Internet. Esto posibilita que puedan ser utilizados desde cualquier lugar y en cualquier momento. La interacción con los usuarios se realiza mediante páginas Web, es decir que estos entornos se presentan al usuario como un sitio Web. Claro está que los EVEAs no son sitios Web cualesquiera, sino que poseen características particulares debido a que están pensados para mediar los procesos de enseñanza y de aprendizaje. Sin embargo, al utilizar la Web como interfaz, es posible contar con una teórica desde donde partir para su análisis, debido a que los lineamientos para un buen diseño Web no deben ser perdidos de vista.

Cuando hacemos referencia al diseño de un sitio Web inmediatamente aparecen conceptos como facilidad de uso, efectividad, eficiencia, facilidad de aprendizaje, satisfacción del usuario, accesibilidad, consistencia, entre otros. Todos estos conceptos están relacionados con la usabilidad del sitio Web. Se

podría decir que la usabilidad de un sitio Web está en relacionada a la facilidad de uso y de aprendizaje [17], [18], [19], y [20]. Sin embargo, esta definición deja de lado otras connotaciones importantes de la usabilidad como lo son la funcionalidad del sitio y el contexto en cual será utilizado [21], [22], y [23]. Puede ocurrir que un sitio Web sea muy fácil de usar, pero no permita al usuario hacer todo lo que desea, es decir, no posea las funcionalidades necesarias. Puede también suceder lo inverso, es decir, que el sitio posea todas las funcionalidades necesarias pero sea muy complicado de usar. Podemos encontrarnos en otras situaciones más complejas como que un sitio sea adecuado, tanto en funcionalidades como en facilidad de uso, para un adolescente con un alto grado de conocimiento en navegación por Internet, una computadora de última generación y una conexión a Internet de alta velocidad, pero se transforme en todo lo contrario para un adulto mayor, con poca experiencia en la Web, con una computadora desactualizada y conexión telefónica. En definitiva, la usabilidad de un sitio Web difícilmente sea la óptima para todos los casos, pero existen una seria de atributos y de reglas que deben ser tenidos en cuenta.

La usabilidad puede ser afrontada como parte del proceso del desarrollo del software o como un atributo del producto final [24]. En el primer caso se trata de una mirada orientada al proceso que es abordada como una disciplina relacionada con la ingeniería de software donde se estudia la forma de diseñar sitios para que los usuarios puedan interactuar con ellos de la forma más fácil, cómoda e intuitiva posible [25]. El segundo caso está orientado al producto final, como un atributo de la calidad del mismo según la perciben los usuarios que hacen uso del sistema.

A menudo se habla del grado de usabilidad de un sistema valorándolo en una escala cualitativa (por ejemplo: bajo, medio, alto) o cuantitativa (por ejemplo: 1 a 10). El grado de usabilidad es una medida empírica y relativa del uso del producto. Es empírica porque se basa en pruebas realizadas en laboratorio u observadas mediante trabajo de campo. Es relativa porque el resultado no es ni bueno ni malo, sino que depende de las metas planteadas o de una comparación con otros sistemas similares.

Hasta no hace mucho tiempo, la usabilidad era una propiedad exclusiva de la presentación de la información, es decir, la capa decorativa del producto. Se creía que encapsulando la capa de presentación y separándola del resto se podía desarrollar la aplicación y obtener altos grados de usabilidad. Sin embargo, este modelo de desarrollo ha fracasado debido a que muchos problemas de usabilidad requieren realizar cambios profundos en la funcionalidad de una aplicación [26]. Dicho de otra manera, además de la usabilidad de una interfaz debe analizarse también el modelo de usuario, el cual está constituido por los objetivos que el usuario quiere alcanzar con sus tareas. Este modelo es el que permite al usuario relacionar sus objetivos con la funcionalidad del sistema [27]. Debido a la creciente complejidad de los sistemas Web, la mayor parte de los aspectos que involucran a la usabilidad no están a la vista, por lo cual para conseguir una buena usabilidad no basta con tener en cuenta solo la capa de presentación, sino que es preciso que se contemple también en el momento de la definición de la funcionalidad de la aplicación [26].

Existen varias definiciones del concepto de usabilidad que son ampliamente referenciadas y forman parte de cualquier estudio sobre el tema. Inclusive algunas de ellas han sido tomadas por organismos internacionales de estandarización como la ISO.

Precisamente el estándar ISO 9241 describe los requisitos ergonómicos para trabajo de oficina con terminales de despliegue visual y explica algunos de los principios básicos subyacentes. En su apartado 11, desarrollado por Bevan, define cómo especificar y evaluar la usabilidad de productos, y aquellos factores que tienen un efecto en la usabilidad desde el punto de vista ergonómico como parte de un sistema de calidad [23]. Este estándar posibilita que evaluadores, consultores, instituciones o empresas no involucrados en el desarrollo del producto puedan medir la usabilidad del mismo. La usabilidad se define como el grado de efectividad, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico [28]. Esta definición deja bien en claro que la usabilidad de un sistema está en directa relación con los usuarios, necesidades y condiciones específicas. Por tanto, la usabilidad no es un atributo inherente al software exclusivamente,

y no puede especificarse independientemente del entorno de uso y de los usuarios concretos que vayan a utilizar el sistema. Para especificar o medir la usabilidad es necesario identificar las metas y descomponer la efectividad, eficiencia y satisfacción, así como los componentes del contexto de uso en sub-componentes con atributos medibles y verificables.

La definición que acabamos de presentar fue tomada por Nielsen con el propósito de ampliar el concepto de usabilidad. Este autor entiende que la usabilidad es un término multidimensional que influye en la aceptación del producto [17]. La define como la medición de la calidad de la experiencia del usuario en interacción con un sistema [17]. Al respecto dice que: "la usabilidad se refiere a la capacidad de un software o sistema interactivo de ser comprendido, aprehendido, usado fácilmente y atractivo para un usuario, en condiciones específicas de uso. También es la efectividad, eficiencia y satisfacción con la que un producto permite alcanzar sus objetivos específicos" [28]. La usabilidad no es algo simple o una propiedad que ofrezca una única cara relacionada con la interfaz de usuario ofrecida, en contraposición tiene múltiples componentes e involucra cinco atributos de calidad más específicos [20]: facilidad de aprendizaje, eficiencia o velocidad de desempeño, recuerdo o retención en el tiempo, tasa de errores por parte de los usuarios y satisfacción.

Por otro lado, Redish elaboró una la definición que se basa en la necesidad de diseñar sitios Web para que los usuarios sean capaces de "encontrar lo que necesitan, entender lo que encuentran y actuar apropiadamente dentro del tiempo y esfuerzo que ellos consideran adecuado para esa tarea" [29]. En este caso el término usabilidad abarca la comprensión de los objetivos de los usuarios, el contexto de su trabajo, y cuál es el conocimiento y la experiencia de que disponen [29]. Es decir, usabilidad significa que la gente que utilice un producto pueda realizar rápida y fácilmente sus tareas.

Para Shackel, la usabilidad está circunscripta al concepto de aceptación, al igual que lo planteado por Nielsen. La aceptación es una función de la percepción de utilidad, uso, agrado y costes. La utilidad hace referencia a la armonía entre las necesidades del usuario y el uso del producto, mientras que el agrado refiere a las evaluaciones afectivas y los costes al ámbito financieros, así como también a las consecuencias sociales y de organización [24]. La usabilidad está definida como la capacidad de un producto de ser utilizado por los seres humanos fácilmente y con eficacia [30]. La eficacia a la que alude Shackel es entendida como sinónimo de la efectividad a la que hacen referencia Bevan y Nielsen. Para este autor, la usabilidad es una característica que no es constante sino relativa a los usuarios, a su entrenamiento y ayuda, a las tareas y a los ambientes. Así, la evaluación es dependiente del contexto [24]. Para que un sistema sea usable tiene que alcanzar niveles definidos en los siguientes criterios operacionales [30]: eficacia, aprendizaje, flexibilidad y actitud. Mientras que Preece [31] señala que el concepto de usabilidad mide la facilidad con la que un sistema puede ser aprendido o usado, incluyendo además la seguridad, efectividad, eficiencia y actitud de los usuarios con respecto al sistema. También Landauer [32] coincide con éste y expresa que el concepto de usabilidad se basa en la estandarización del comportamiento y funcionamiento de las interfaces sumado a la facilidad de uso y capacidad de aprendizaje [33].

Las definiciones presentadas anteriormente poseen atributos comunes entre sí, como también aspectos que las distinguen.

	Atributos					
Autor	Efectividad	Eficiencia	Satisfacción	Contexto	Aprendizaje	Flexibilidad
Bevan	●	●	●	●		
Nielsen	●	●	●	●	●	
Redish	●	●	●	●		
Shackel	●		●	●	●	●

En todos los casos está presente la efectividad, la satisfacción por parte del usuario y el contexto de uso como elementos a considerar. Shackel incorpora el concepto de flexibilidad, pero por la manera de definirla también incluye al contexto de uso.
Los atributos considerados en las definiciones presentadas anteriormente no son los únicos. Existen otros como la consistencia y el estilo comunicativo. También es muy común que la efectividad sea expresada en función de tres sub-atributos: facilidad de aprendizaje, tasa de errores y recuerdo en el tiempo.
Sin embargo, la usabilidad de un sistema no es una simple adición del valor de estos atributos, sino que se define para cada sistema como un nivel a alcanzar. Cada uno de los atributos puede descomponerse a su vez para conseguir una mayor precisión en los aspectos de usabilidad en los que se quiere poner mayor énfasis. La eficiencia, por ejemplo, puede analizarse en función del rendimiento en uso normal y el uso de opciones avanzadas, mientras que la primera impresión que tiene un usuario sobre el sistema podría ser un sub-atributo de satisfacción. Algunos de estos atributos no contribuyen a la usabilidad del sistema en la misma dirección, pudiendo ocurrir que el aumento de uno de ellos tenga como efecto la disminución de otro. Por ejemplo, si se desea un sistema con alta facilidad de aprendizaje y eficiencia, es conveniente no sobrecargar la interfaz y agregar el uso de teclas rápidas para ejecutar operaciones de uso habitual [34]. En otros casos un atributo puede ser mucho más relevante que otro y tener una ponderación acorde a la actividad que se quiera realizar con el sistema, de las características de la audiencia objetivo y de las circunstancias en las cuales se usará el software [35].
Es decir, la usabilidad encuentra relación con otros conceptos íntimamente ligados al usuario tales como: quiénes son los usuarios, cuáles sus conocimientos, qué pueden aprender, qué quieren o necesitan hacer, cuál es la formación general, cuál es el contexto en el que el usuario está trabajando, qué debe dejarse a la máquina y qué al usuario, el tipo de producto del que se trata, el tipo de usuario que hace uso del producto y las prestaciones que logren los usuarios haciendo uso del producto [20][25]. Esto hace que el papel del usuario en todo proceso de evaluación sea vital, ya que sin usuario no hay evaluación [33].

Heurísticas

La heurística trata de aplicar normas convencionales a la interacción entre el hombre y la máquina. Su objetivo es reducir la brecha entre el usuario y el sistema para favorecer el logro de los objetivos. Se habla de heurísticas en el sentido de que se trata de directrices o reglas realizadas por expertos que pueden ser utilizadas como guía para los diseñadores, como ayuda a los evaluadores, para explicar problemas de usabilidad observados y para dar pautas de porqué los usuarios cometen ciertos errores [36]. No proveen una definición concreta del concepto de usabilidad sino que lo definen enfocando las metas del diseño [24]. Como podremos observar, las heurísticas guardan relación con los atributos de usabilidad. Ellas pueden tener diferente nivel de especificidad ya que podemos encontrar orientaciones genéricas para el diseño de todo un sitio, como lo es la adecuación a ciertos estándares, o sugerencias tan concretas como cual debería ser la longitud del texto de un enlace. Las heurísticas más relevantes se deben a expertos en la materia como Schneiderman, Nielsen, Instone, Constantine, Mayhew, Norman y Tognazzini.
Como podemos observar éstas heurísticas poseen similitudes y diferencias. En la siguiente tabla se especifican las más importantes.

	Schneiderman	**Nielsen e Instone**	**Constantine**	**Mayhew**	**Norman**	**Tognazzini**
Consistencia	●	●	●		●	●
Visibilidad		●	●		●	●
Retroalimentación	●		●	●	●	
Control por parte del	●	●		●	●	●

usuario						
Reversibilidad	●	●	●			●
Manejo del error	●	●		●		
Reutilización	●	●	●			
Simplicidad			●	●		●
Metáforas		●		●	●	●
Presentación visual		●			●	●
Productividad	●	●				
Seguridad				●		●
Ayuda y documentación		●				

4. La evaluación de la usabilidad

La principal actividad en el proceso de usabilidad es la evaluación [34]. La evaluación comprende un conjunto de métodos y técnicas que analizan la usabilidad de la versión final de un sistema o a lo largo de sus diferentes etapas de desarrollo. Evaluar la usabilidad de un sitio Web es extremadamente útil ya que permite descubrir errores y aciertos de diseño [22]. La evaluación de la usabilidad ayuda a determinar cual es el nivel actual de la aplicación, y si de hecho el diseño elegido realmente funciona. Los datos que se recaban mediante la observación del desempeño del usuario frente a la aplicación son muy valiosos puesto que ayudan a detectar posibles falencias del sistema [35].

Varios autores coinciden en que la usabilidad puede ser medida cuantitativamente siempre que se utilicen las variables y métodos adecuados [17] [44]. Por ejemplo, se podría medir la usabilidad a través del tiempo empleado en localizar una función o menú determinado, el número de pasos que se necesitan para llevar a cabo una función, el tiempo empleado en realizar una tarea específica o el tiempo que tarda un usuario en aprender a dominar el sistema. Aunque sin perder de vista que no todos los atributos de la usabilidad se evalúan de la misma manera, en algunos casos será necesario utilizar tests, expertos, etc.

Por ejemplo, para especificar o medir la usabilidad desde la definición propuesta por la ISO 9241 es necesario descomponer los factores de eficacia, eficiencia y satisfacción, así como los aspectos relevantes del contexto de uso, en atributos que puedan ser medidos y verificados. La especificación o medida de la usabilidad puede entonces expresarse por el grado en que se logran los objetivos que se pretenden del sistema (eficacia), los recursos que han de gastarse para lograr dichos objetivos (eficiencia) y el grado en que los usuarios encuentran aceptable el conjunto del sistema (satisfacción). El contexto de uso integra las características o atributos relevantes de la tarea, usuario, equipamiento y medio ambiente que tienen influencia en la usabilidad [21].

Existen distintos métodos para evaluar la usabilidad de un sistema que intentan medir diferentes aspectos de la misma [45] [23]. Su uso depende de variables tales como grado de implicación del usuario, escenarios de tarea, costo, disponibilidad de tiempo, personal calificado para interpretar los datos, entre otros factores [35]. Inclusive es altamente beneficioso no utilizar un solo método de evaluación, sino varios que se complementen mutuamente y permitan observar las características del producto desde diferentes puntos de vista. Además, cada método puede constar de diferentes técnicas. Estos métodos pueden ser clasificados por numerosos criterios ya que no existe un acuerdo común que permita agruparlos. Incluso muchas veces los criterios no permiten una clasificación del todo clara debido a que no son totalmente independientes uno de otros [46]. Uno de los criterios más aceptados es aquel que los clasifica en métodos de inspección, métodos de indagación y métodos de test.

Los métodos de inspección son llevados a cabo por expertos que realizan un examen de la interfaz y emiten un juicio sobre la misma. Generalmente uno de los participantes actúa como moderador, destacando las fortalezas y las debilidades de la aplicación. Dentro de estos métodos encontramos a la

evaluación heurística, inspecciones formales, inspección de características, inspección de consistencia, inspección de estándares, paseo cognitivo y listas de comprobación. Los más destacados son la inspección formal, la evaluación heurística y el paseo cognitivo.
Los métodos de indagación, también llamados de investigación, consisten en hablar y/o observar al usuario usando el artefacto o producto. Dentro de los métodos de indagación podemos encontrar diversas formas de aproximación al usuario: contextual (en el contexto, estudio etnográfico/observación de campo), por grupos (orientados, de debate), individual (entrevistas, encuestas, cuestionarios), sesiones guiadas, registro por el usuario, captura de pantallas y observación experta. Por su facilidad de administración y rapidez de evaluación, los cuestionarios son uno de los métodos más utilizados seguido de las entrevistas y los registros por el usuario.
Los métodos de test, también llamados empíricos, se basan en que usuarios representativos trabajen en tareas utilizando el sistema o un prototipo y los evaluadores utilicen los resultados para ver cómo la interfaz soporta tareas y usuarios. El objetivo del test es adaptar el software a los estilos de trabajo reales de los usuarios, en lugar de forzar a los usuarios a adaptar sus estilos de trabajo al software. Es una de las mejores formas de extraer información sobre el nivel de usabilidad de un sitio Web. Se trata de la más importante ya que proporciona información de su uso real observando a los usuarios finales cómo utilizan el sitio para realizar determinadas tareas, siendo la única forma válida de recoger datos fiables de usabilidad [47]. Consiste en observar cómo un usuario intenta llevar a cabo una tarea dada sobre el sitio Web a evaluar, analizando los problemas con los que se encuentra. El test busca obtener información específica acerca de un diseño, registra los problemas que encuentran los usuarios cuando tratan de realizar las acciones concretas propuestas por el evaluador, así como los tiempos empleados en cada una de ellas. Estos test están compuestos por la realización de medidas de desempeño, protocolo de preguntas, protocolo del pensamiento en voz alta, protocolo del descubrimiento conjunto, tutorado, seguimiento visual y ordenación de tarjetas.

5. Hacia un modelo de evaluación centrado en la usabilidad

En esta sección presentaremos algunas ideas abordadas como parte del trabajo de investigación que estamos realizando con el fin de proponer un modelo de evaluación que esté centrado en la usabilidad.
Como hemos mencionado, la participación del usuario en el proceso de evaluación es imprescindible para la obtención del grado de usabilidad de un software. Esto posibilita que pueden involucrarse personas ajenas al desarrollo, tanto para evaluar productos en desarrollo, productos terminados o nuevas versiones. Por lo tanto, es necesario que las herramientas de evaluación sean participativas y cuenten con la posibilidad cierta de incorporar a los usuarios finales en el proceso. Esto no implica que sean los únicos actores involucrados, sino que tal vez sea necesaria la intervención de observadores o personas expertas en el dominio o herramientas de automatización [48]. Para lograr que el modelo sea efectivamente utilizado, el proceso de evaluación y la interpretación los resultados debe poder hacerse de la forma más rápida y menos costosa posible.
La usabilidad no puede determinarse evaluando un producto de manera aislada sino que debe determinarse en función del tipo de producto, el conjunto de usuarios y el contexto de uso. Es una medida relativa, y por lo tanto difícilmente existan reglas de usabilidad absolutas y aplicables por igual en todas las situaciones [49].
Los modelos de evaluación existentes analizan los EVEAs poniendo el foco sobre las funcionalidades que ofrecen. El análisis de las funcionalidades ideales no tiene demasiado en cuenta la usabilidad. Por ejemplo, se puede contar con una herramienta comunicación como el foro, pero se debe analizar la usabilidad de esta herramienta por parte de docentes y alumnos involucrados.
Los modelos de evaluación de EVEA fueron pensados cuando las características diferían bastante entre un entorno y otro. En la actualidad, los entornos de mayor importancia poseen un núcleo común de herramientas. Donde todavía resta camino por recorrer es en cuestiones ligadas a la usabilidad,

posiblemente porque son más difíciles de medir, como por ejemplo la flexibilidad para adecuar, cada entorno y sus herramientas, a distintos situaciones, contextos o modelos de enseñanza y de aprendizaje. En el mejor de los casos, la usabilidad está restringida a la facilidad de uso y el cumplimiento de ciertos estándares.

Las definiciones y heurísticas de usabilidad presentan una idea sobre las consideraciones que hay que tener a los efectos de alcanzar buenos niveles de usabilidad en los sitios Web. Sin embargo, no existe un modelo de evaluación de EVEAs que integre ambas cuestiones. Para superar estos inconvenientes es necesario contar con un modelo de evaluación específico. Un modelo centrado en la usabilidad debería nutrirse de los ya existentes debido a que la evaluación de la funcionalidad es parte importante, pero debe tener en cuenta a docentes y alumnos trabajando en el entorno con la funcionalidad disponible.

Otro aspecto de gran relevancia para alcanzar altos niveles de usabilidad es la accesibilidad. Para que un entorno sea usable, entre otras cosas, debe ser accesible. Dicho de otra manera, la accesibilidad es condición necesaria para la usabilidad. Por ejemplo, muchos usuarios con computadoras viejas, navegadores desactualizados, o conexiones lentas, estarán impedidos de acceder si la página principal de acceso a las distintas herramientas de un curso está realizada exclusivamente con tecnología flash.

A continuación presentaremos las ideas generales o punto de partida para la búsqueda de un nuevo modelo. De acuerdo a lo analizado, una estrategia sería contar con cuatro niveles de evaluación:

- El primer nivel estaría destinado a analizar características funcionales y de interfaz. Se considerarían aquí cuestiones relacionadas con las facilidades para la organización académica (cursos, grupos de cursos, carreras, comunidades, etc.) y flexibilidad pedagógica. Participarían especialistas en educación y en tecnología. Se tomarían como base los criterios recurrentes en los modelos de evaluación de EVEAs analizados anteriormente.
- En el segundo nivel participarían expertos para realizar una evaluación heurística y un recorrido cognitivo: estos dos métodos de inspección podrían ser conjugados en uno solo. En el recorrido cognitivo el especialista transita un escenario de tareas determinado como lo haría un usuario tipo. Es decir, combina el recorrido del software con un modelo cognitivo de aprendizaje por exploración. La evaluación heurística es una variante de la inspección formal donde los especialistas analizan que cada elemento de la interfaz de usuario siga las heurísticas de usabilidad establecidos y recorren meticulosamente las tareas con los propósitos y objetivos de los usuarios en mente, poniendo énfasis en el hallazgo de errores. Los dos métodos realizan recorridos desde el punto de vista del usuario, pero los paseos cognitivos ponen el énfasis en la teoría cognitiva mientras que la evaluación heurística lo hace más en las heurísticas y el hallazgo de errores.
- En el tercer nivel se realizaría un test de usuario, del tipo de protocolo de expresión del usuario en base a preguntas (variante del de pensamiento en voz alta): participaría un evaluador y algunos usuarios finales realizando una tarea como parte de un escenario tipo. Es importante que se involucren los diferentes roles de usuarios que participan en el uso de un EVEA. A medida que interactúa con el entorno el usuario debería expresar en voz alta sus pensamientos, sensaciones y opiniones. El evaluador debería captar y registrar las impresiones y sensaciones que el usuario manifiesta oralmente y realizar preguntas directas.
- El cuarto nivel involucraría al método de indagación por cuestionarios: también participaría un evaluador y usuarios finales, pero no estarían cara a cara sino que el evaluador sería el encargado de suministrar el cuestionario y procesar los resultados. Los cuestionarios estarían formados por una lista de preguntas asociadas a ciertas tareas que deben realizar los usuarios. El usuario debe realizar las tareas en soledad, contestar las preguntas y enviar el cuestionario al evaluador. Los cuestionarios podrán ser interactivos y estar colocados directamente dentro del mismo entorno a ser evaluado.

Los niveles plantean ciertos grados de contención del usuario, siendo el primer nivel el de mayor contención. La transición de un nivel a otro estaría dada por resultados aceptables en las evaluaciones. Es decir que, en el caso de productos en desarrollo, la usabilidad debería mejorar de un nivel a otro, producto de que los errores más importantes han sido detectados y corregidos.
El primer nivel es el encargado de realizar una evaluación del entorno en general, mientras que los restantes se sitúan en un contexto del uso particular, por ejemplo un curso. El abordaje general del primer nivel tendría como objetivo analizar aspectos tales como:

- Flexibilidad del sistema para adaptar sus funcionalidades a diferentes situaciones, contextos o modelos de enseñanza y de aprendizaje. La flexibilidad buscada puede ser tanto desde el punto de vista de la interfaz como del modelo de dominio. En el primer caso nos referimos a la posibilidad de adecuar el aspecto visual del sistema de acuerdo a las necesidades institucionales: banners, logos, colores, tipografías, etc. En el segundo caso consideramos las facilidades del sistema para trabajar en el marco de un curso con pocos o muchos alumnos, con un docente o varios. Además, la posibilidad de seleccionar las herramientas o recursos que serán utilizados, de definir los roles que tendrá cada integrante dentro de curso (alumno, estudiante, docente, tutor, responsable, profesor, etc.) y la forma de utilizar cada herramienta por parte de cada tipo de usuario (no disponible, solo lectura, modificación, etc.).
- Nivel de experticia que necesita el usuario de acuerdo a su rol. No es lo mismo el nivel de conocimiento que necesita un usuario que solo debe navegar por el sitio, descargar un material digital y opinar en un foro, que aquel usuario que debe hacerse cargo de la gestión de usuarios, materiales, grupos y foros, y la generación de evaluaciones on-line. Cuando decimos gestión nos referimos a los procesos de administración en general (altas, bajas, modificaciones, etc.).

En los restantes niveles el objetivo sería analizar las herramientas que posee el sistema y la forma de utilizarlas en función de los objetivos del usuario, tal como se mencionó, y considerando las heurísticas de usabilidad presentadas en la correspondiente sección.

Es importante aclarar que este modelo de evaluación que se está proponiendo y que se continúa refinando y delineando será aplicado para evaluar el entorno SIAT [50] utilizado en el marco de la Universidad Nacional de Río Cuarto. Se seleccionarán docentes y alumnos, así como diseñadores del producto y otros expertos que participarán en la evaluación.

Los instrumentos específicos se diseñarán oportunamente para llevar adelante la evaluación.

6. Conclusiones

En este artículo se ha realizado una presentación del concepto de entornos virtuales de enseñanza y aprendizaje y una revisión de su evolución.
Hemos realizado, también, un recorte de los modelos de evaluación de EVEAs presentados por diferentes autores.
Se ha delineado el concepto de usabilidad y el de heurísticas, también analizando la posición de diferentes autores reconocidos en la temática, y generando una tabla comparativa que ayude a visualizar rápidamente aspectos comunes y de diferenciación abordados.
Esto ha sido el punto de partida para la propuesta que están desarrollando los autores de este trabajo, de un modelo de evaluación de EVEAs que de una real importancia al concepto de usabilidad.

Este modelo continúa en desarrollo y estará siendo aplicado a un caso específico de estudio, que dará a lugar a una retroalimentación por parte de los involucrados en la evaluación para perfeccionar el modelo.

Referencias

[1] Barberá E. La incógnita de la Educación a Distancia. *Cuaderno de Educación. ICE - HORSORI. Barcelona, España.* 2001.
[2] Sigales C. El potencial interactivo de los entornos virtuales de enseñanza y aprendizaje en la educación a distancia. *X encuentro Internacional de Educación a Distancia. Guadalajara, México.* 2001.
[3] Fainholc B. La interactividad en la educación a distancia. *1ª ed. Buenos Aires. Editorial Paidós*. 1999.
[4] Torres Toro S. y Ortega Carrillo J.A. Indicadores de calidad en las plataformas de formación virtual: una aproximación sistemática. *Etic@ Net, 1. http://www.ugr.es/~sevimeco/revistaeticanet/ Granada, España.* 2003.
[5] Adell J. Selección de un entorno virtual de enseñanza/aprendizaje de código fuente abierto para la Universitat Jaume I. *Centre d'Educació i Noves Tecnologies de la UJI con la colaboración del Servei d'Informàtica y del Gabinet Tècnic del Rectorat*. 2004.
[6] Horton S. Web teaching guide. *New Haven: Yale University Press*. 2000.
[7] Baumgartner P, Häfele H., et al. E-Learning Praxishandbuch: Auswahl von Lernplattformen. *Marktübersicht - Funktionen - Fachbegriffe. Innsbruck-Wien, StudienVerlag*. 2002.
[8] Hernández Schäfer L.E. Estado actual y futuro de las plataformas e-learning: estándares y especificaciones IMS. *Universidad Católica del Norte, Chile*. 2002
[9] López Carrasco M.A. Plataformas Virtuales de Aprendizaje. Athenea. Puebla, México. *www.athenea.com.mx/swf/plataformas_virtuales.pdf*. 2003.
[10] Rubio M.J. Enfoques y modelos de evaluación del e-learning. *Revista ELectrónica de Investigación y Evaluación Educativa, v. 9, n. 2. http://www.uv.es/RELIEVE/v9n2/RELIEVEv9n2_1.htm.* 2003.
[11] Pallof R.M. and Pratt K. Lessons from the cyberspace classroom. *San Francisco: Jossey-Bass*. 2001.
[12] Gabinete de Tele-Educación. Plataformas de tele-enseñanza. *Universidad Politécnica de Madrid. http://www.gate.upm.es*. 2002.
[13] Zapata Ros M. Evaluación de un Sistema de Gestión del Aprendizaje. *www.um.es/ead/red/9/eval_SGA_1.pdf*. 2003.
[14] Marquès Graels, P. Criterios de calidad para los sistemas de teleformación. *http://dewey.uab.es/pmarques/stfcalid.htm.* 2001
[15] Grimes G.T and Whitmyer C. Comparative Features Analysis of Leading Course Management Software. *The University of the Future, LLC*. 2000.
[16] COL LMS Open Source Report. *The Commonwealth of Learning, 3waynet. http://www.developmentgateway.org/download/201768/*. 2003.
[17] Nielsen J. Iterative User-Interface Design. *IEEE Computer.* 1993.
[18] Schneiderman B. Designing the User Interface: Strategies for Effective Human-Computer Interaction. *Third edition, Addison Wesley Iberoamericana.*1998.
[19] Manchón E. ¿Qué es la usabilidad? *http://www.alzado.org/articulo.php?id_art=39.* 2003.
[20] Nielsen J. Usability 101: Introduction to Usability. *http://www.useit.com/alertbox/20030825.html*. 2003.
[21] Sanz J.A. Manual de normas técnicas para el diseño ergonómico de puestos con pantallas de visualización. *2ª Edición. ISO-9241. http://www.mtas.es/insht/practice/pvd.htm*. 2001.

[22] Montero F. Usabilidad: ¿Qué?¿Cómo?¿Cuándo? ¿Dónde? ¿Para qué?. *Universidad de Castilla-La Mancha. España. http://www.dsi.uclm.es/personal/victor/mipagina/index.htm*. 2006
[23] Cañas J., Granollers T. y Vidal J. Diseño de Sistemas Interactivos Centrados en el Usuario. *Editorial UOC. ISBN: 8497883209. http://griho.udl.es/mpiua/index.htm*. 2005.
[24] Keinonen T. Usabilidad de los productos interactivos. *http://www2.uiah.fi/projects/metodi/258.htm*. 2003.
[25] Hassan Montero Y. Introducción a la Usabilidad. *No Solo Usabilidad journal, 1. ISSN 1886-8592. Universidad de Granada. http://www.nosolousabilidad.com/articulos/introduccion_usabilidad.htm. España*. 2002.
[26] Casanovas J. Usabilidad y arquitectura del software. *http://www.alzado.org/articulo.php?id_art=355*. 2004
[27] Berry D. The user experience: The iceberg analogy of usability. *http://www-128.ibm.com/developerworks/library/w-berry/*. 2000
[28] ISO 9241-11. Ergonomic requirements for office work with visual display terminals. Guidance on Usability. *ISO.* 1998.
[28] Nielsen J. Usabilidad: Diseño de sitios web. *Prentice Hall, ISBN: 84-205-3008-5.* 2001.
[29] Dumas, J.S. and Redish, J.C. A practical guide to usability testing. *Norwood: Ablex Publishing Corporation.* 1993.
[30] Shackel B. Usability – context, framework, design and evaluation. *En Human Factors for Informatics Usability. Cambridge University Press*, 1991.
[31] Benyon D., Carey T., Holland S., Preece J., Rogers Y. and Sharp H. Human-Computer Interaction. *Addison Wesley*, 1994.
[32] Landauer T.K. The Trouble with Computers: Usefulness, Usability and Productivity. *Massachusetts: Massachusetts Institute of Technology*, 1995.
[33] Catalán Vega, M. Metodologías de evaluación de interfaces gráficas de usuario. *http://eprints.rclis.org/archive/00004718/*. 2000.
[34] Constantine L., Ferré X., Juristo N. and Windl H. Usability Basics for Software Developers. *IEEE Software, vol 18 (11).* 2001.
[35] Baeza Yates R. y Rivera Loaiza C. Ubicuidad y Usabilidad en la Web. *Revista Colombiana de Computación.* 2003.
[36] Gonzalez M.P., Lorés J. y Pascual A. Evaluación heurística. *Universitat de Lleida. http://griho.udl.es/ipo/pdf/15-Evaluacion-Heuristica.pdf*. 2006.
[37] Nielsen J. Ten Usability Heuristics. *Useit.com http://www.useit.com/papers/heuristic/heuristic_list.html*. 1990.
[38] Instone K. Site Usability Heuristics for the Web. *Webreview.com http://www.webreview.com/1997/10_10/strategists/10_10_97_2.shtml*. 1997.
[39] Constantine L. and Lockwood L. Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. *Addison-Wesley, Nueva York, NY*, 1999.
[40] Mayhew D. The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design. *The Morgan Kaufmann Series in Interactive Technologies.* 1999.
[41] Norman D. Psicología de los objetos cotidianos. *Nerea. ISBN: 84-89569-18-5.* 1998.
[42] Norman D. Emotional Design: Why we love (or hate) everyday things. *Basic Books.* 2004.
[43]Tognazzini B. First Principles of Interaction Design. *Ask Tog*. *http://www.asktog.com/basics/firstPrinciples.html.* 2003
[44] Mandel T. Elements of User Interface Design. *New York: John Wiley & Sons*, 1997.
[45] Cobo Romaní C. Organización de la información y su impacto en la usabilidad de las tecnologías interactivas. *Tesis doctoral. Facultat de Ciències de la Comunicació, Universitat Autònoma de Barcelona, Barcelona, España*. 2005

[46] Rodeiro, J. Representación y Análisis de la componente visual de la interfaz de usuario. *Tesis doctoral. Universidad de Vigo, Vigo, España.* 2001.
[47] Nielsen J. Voodoo Usability. *Alertbox. http://www.useit.com/alertbox/991212.html.* 1999.
[48] Gonzalez Rodríguez J. y Olsina L. Hacia la Medición de Calidad en Uso Web. *VI JISBD´2001 - Workshop de Web Engineering. Almagro, España.* 2001.
[49] Marcos Mora M.C. y Rovira Fontanals C. Evaluación de la usabilidad en sistemas de información web municipales: metodología de análisis y desarrollo. *En: Gascon, J., Burguillod F. y Pons A. (ed.). La dimensión humana de la organización del conocimiento, 7º Congreso del capítulo español de Isko. Universitat de Barcelona, Barcelona, España.* 2005.
[50] Ferreira Szpiniak A y Thüer S. Características, potenciales y uso de la plataforma de educación a distancia SIAT. *1a. ed. ISBN 950-665-276-7. Universidad Nacional de Río Cuarto, Córdoba, Argentina.* 2004.

Hearing The World: Um Ambiente de Apoio à Alfabetização e Comunicação de Deficientes Auditivos e Surdos

Thatiane de Oliveira Rosa
Complexo de Informática, Centro Universitário Luterano de Palmas (CEULP/ULBRA)
Palmas, Tocantins, Brasil
thatiane@ulbra-to.br

Danielle Corrêa Ribeiro
Complexo de Informática, Centro Universitário Luterano de Palmas (CEULP/ULBRA)
Palmas, Tocantins, Brasil
danielle@ulbra-to.br

Elizabeth Maria Martinho da Silva
Complexo de Informática, Centro Universitário Luterano de Palmas (CEULP/ULBRA)
Palmas, Tocantins, Brasil
elizabeth@ulbra-to.br

Cristina D'Ornellas Filipakis
Complexo de Informática, Centro Universitário Luterano de Palmas (CEULP/ULBRA)
Palmas, Tocantins, Brasil
cristina@ulbra-to.br

Abstract

A significant parcel of the Brazilian population is carrying of auditory deficiency and deafness. A notable percentage of this population belongs to school age, but few are inserted in this circuit. Thus, it is evidenced that so low amount of students with auditory deficiency and deafness occurs for the fact of Libras (Língua Brasileira de Sinais – Signals Brazilian Language) is not a study object, even so many people dominates it. On the other hand, the listeners do not know the signals' language, what makes difficult their communication with auditory deficient and deaf. This manner, the development of an educative environment was thought, aiming at to assist auditory deficient and deaf as much in their reading and writing's process as well as in their communication with listeners throughout the life.

Key-words: auditory deficiency and deafness, Libras, communication

Resumo

Uma significativa parcela da população brasileira é portadora de deficiência auditiva e surdez. Uma porcentagem notável dessa população encontra-se em idade escolar, porém poucos estão inseridos neste âmbito. Desta forma, constata-se que tão baixa quantidade de estudantes com deficiência auditiva e surdez dá-se pelo fato de Libras (Língua Brasileira de Sinais) não ser um objeto de estudo, embora várias pessoas a dominem. Por outro lado, os ouvintes não conhecem a linguagem de sinais, o que torna sua comunicação com deficientes auditivos e surdos complexa. Com isso, pensou-se no desenvolvimento de um ambiente educativo que visa auxiliar deficientes auditivos e surdos tanto no seu processo de alfabetização como também em sua comunicação com ouvintes ao longo de sua vida.

Palavras-chave: deficiência auditiva e surdez, Libras, comunicação

1. INTRODUÇÃO

Conforme o Censo 2000 do IBGE [1], aproximadamente 5.790.000 pessoas da população brasileira são portadoras de deficiência auditiva, em diferentes graus, sendo deste total cerca de 160.000 surdas. Outro fato relevante é que mais de 406.000 dos brasileiros com algum grau de deficiência auditiva estão em idade escolar, sendo que por volta de 13% estavam matriculados na educação básica em 2003, ou seja, 87% dessas crianças estão fora da escola, como divulga o Censo Escolar 2004. Ainda, pouco mais de 2.000 estudantes cursam o ensino médio e, somente 300 estão inseridos no Ensino Superior [2].

Assim, pode-se notar que um dos motivos de tão baixa porcentagem de estudantes com deficiência auditiva é o fato de não haver aulas na Língua Brasileira de Sinais ([1]Libras), tornando-se a barreira central da inserção de tal linguagem no sistema escolar, embora se estime que 60% das pessoas portadoras de tal deficiência conhecem Libras. Por outro lado, os ouvintes não entendem a linguagem de sinais, o que torna sua comunicação com os surdos complexa. Portanto, o processo de aprendizagem e a inclusão social dos surdos mostram-se trabalhosas. Visando amenizar tal problema, sancionou-se a Lei nº 10.436, de 24 de abril de 2002, que reconhece como meio legal de comunicação e expressão a Língua Brasileira de Sinais e outros recursos de expressão a ela associados [3].

Através de pesquisas de campo realizadas em escolas públicas e privadas do Estado do Tocantins, sendo algumas especializadas no ensino a pessoas com deficiência auditiva e surdez, constatou-se a visível falta de estrutura que há no que diz respeito ao atendimento de tais pessoas, fato que também ocorre na maioria das instituições brasileiras de ensino. Essa falta de estrutura está relacionada à capacitação de profissionais na interação e adoção de metodologias pedagógicas não clássicas, uma vez que, como apresentado, Libras não é comumente utilizada e, ainda, tais profissionais deparam-se com situações não antes enfrentadas em seu dia-a-dia, o que muitas vezes causa um sentimento de receio e desafio. Além disso, são poucas as ferramentas tecnológicas que tentam auxiliar nesse processo de adaptação educacional.

Outro problema de fundamental importância é o preconceito existente para com os deficientes auditivos e surdos, ainda hoje presente em uma sociedade antropocêntrica regida por modismos que determinam, além de vestimentas, condutas. Na maioria das vezes, o que ocasiona tal discriminação é a falta de informação sobre o assunto.

Como tentativa parcial de solucionar os problemas relatados, no dia 22 de dezembro de 2005 foi legislado o Decreto nº 5.626 que, no Capítulo IV, visa garantir o uso e a difusão da Libras e da Língua Portuguesa para o acesso das pessoas surdas à educação. Segundo o Artigo 14, "As instituições federais de ensino devem garantir, obrigatoriamente, às pessoas surdas, acesso à comunicação, à informação e à educação nos processos seletivos, nas atividades e nos conteúdos curriculares desenvolvidos em todos os níveis, etapas e modalidades de educação, desde a educação infantil até a superior" [4]. Ainda, no § 1º do referente Artigo, "Para garantir o atendimento educacional especializado e o acesso previsto no **caput**, as instituições federais de ensino devem: ofertar, obrigatoriamente, desde a educação infantil, o ensino da Libras e também da Língua Portuguesa, como segunda língua para os alunos surdos; prover as escolas com professores de Libras ou instrutor de Libras, tradutor e intérprete de Libras-Língua Portuguesa, professor para o ensino de Língua Portuguesa como segunda língua para pessoas surdas, professor regente de classe

[1]Libras – Forma de comunicação e expressão, em que o sistema lingüístico de natureza visual-motora, com estrutura gramatical própria, constitui um sistema lingüístico de transmissão de idéias e fatos, oriundos de comunidades de pessoas surdas do Brasil [4].

com conhecimento a cerca da singularidade lingüística manifestada pelos alunos surdos; garantir o atendimento às necessidades educacionais especiais de alunos surdos, desde a educação infantil, nas salas de aula e, também, em salas de recursos, em turno contrário ao da escolarização; apoiar, na comunidade escolar, o uso e a difusão de Libras entre professores, alunos, funcionários, direção da escola e familiares, inclusive por meio da oferta de cursos; disponibilizar equipamentos, acesso às novas tecnologias de informação e comunicação, bem como recursos didáticos para apoiar a educação de alunos surdos ou com deficiência auditiva" [4].

2. MATERIAIS E MÉTODOS

2.1. Metodologia Adotada

O desenvolvimento do sistema teve início através de pesquisas de campo realizadas tanto em escolas públicas quanto privadas, sendo que dentre estas podem ser encontradas tanto instituições tradicionais quanto especializadas no ensino de deficientes auditivos e surdos. Nas várias instituições visitadas, foram selecionadas uma tradicional e outra especializada, localizadas nos municípios de Paraíso do Tocantins e Palmas, no Estado do Tocantins, Brasil. Na instituição tradicional, obtiveram-se relatos de vários profissionais, os quais assumiram não possuírem métodos didáticos apropriados para aplicar no processo de aprendizado de estudantes portadores de deficiência auditiva e surdez, como também um preparo no que diz respeito à postura que é preciso assumir na interação com esses estudantes. Nesta instituição, estudantes portadores de tal deficiência encontram-se inseridos em salas de aula juntamente com alunos ouvintes, porém não possuem uma participação efetiva durante as aulas. Estes ficam isolados, realizando atividades diferentes dos demais alunos da sala de aula. Já na instituição especializada escolhida, aulas exclusivas são oferecidas, na qual os alunos realizam atividades que abordam tanto a Língua Portuguesa quanto Libras. Vale ressaltar que nas instituições visitadas constatou-se que os alunos que apresentam deficiência auditiva e surdez obtêm um melhor desempenho quando estes se encontram utilizando computadores, apresentando uma visível facilidade no manuseio destes. Todos os fatos citados foram comprovados por meio de entrevistas com educadores e análise de comportamento em sala durante as aulas em ambas as instituições de ensino, realizadas pelas desenvolvedoras do projeto.

Foram também realizadas pesquisas bibliográficas, tanto em meio eletrônico quanto em livros e revistas. Estas possibilitaram a aquisição de conhecimento a respeito de censos, estatísticas, leis e decretos; formas pedagógicas de ensino; instituições que trabalham exclusivamente com deficientes auditivos e surdos; situações vividas por deficientes, os quais relatam as dificuldades por eles enfrentadas em seu dia-a-dia; jogos pedagógicos; e sistemas que tratam da tradução da Língua Portuguesa para Libras.

A partir de então, procurou-se por profissionais da educação, de fundamental importância para o levantamento de requisitos do sistema, e deu-se a modelagem do sistema e sua implementação de forma iterativa. Foram procurados também profissionais da área tecnológica, os quais auxiliaram na seleção de tecnologias destinadas à captação de som. Tais profissionais ainda auxiliaram na escolha de periféricos e dispositivos móveis que suportem o sistema e, ao mesmo tempo, de um custo financeiro acessível à maioria das instituições brasileiras.

2.2. Materiais

Quanto às tecnologias, para a modelagem do sistema utilizou-se a UML (*Unified Modeling Language*), por ser uma linguagem-padrão para a estruturação de projetos de *software*.

O sistema baseia-se em .NET por ser uma plataforma de desenvolvimento de aplicações rica em recursos e que, ao mesmo tempo, é um ambiente gerenciado e de execução protegida. A linguagem C# foi escolhida para a implementação do sistema, por esta ser compatível com a plataforma adotada, além de ser simples, moderna, orientada a objetos, fortemente tipada e robusta. As características de tal linguagem possibilitam a rápida construção de soluções. Os módulos do sistema foram desenvolvidos utilizando a ferramenta Microsoft Visual Studio 2005 [5], que consiste em um ambiente de desenvolvimento abrangente, que possibilita a construção de aplicações com multicamadas de alta performance e se caracteriza como um ambiente produtivo, possibilitando a criação de uma grande variedade de soluções baseadas no Windows, Web e Dispositivos Móveis. No que se refere à implementação do módulo de comunicação, testes foram realizados através do emulador do próprio Visual Studio 2005. Além disso, vale ressaltar que tal ferramenta suporta a plataforma .NET e possibilita a codificação da linguagem C#.

Para a codificação da base de dados do sistema, adotou-se o SQL (*Structured Query Language*), por esta ser uma linguagem padrão para realizar operações de acesso a Banco de Dados. Esta linguagem é independente de *hardware* e *software*, além de ser simples, intuitiva e permitir o tratamento e manipulação de dados relacionais. Quanto ao SGBD (Sistema de Gerenciamento de Banco de Dados), optou-se pela utilização do Microsoft SQL Server 2005 [6], por possuir uma estreita integração com o Visual Studio 2005, ser uma plataforma de base de dados abrangente, segura e confiável.

Utiliza-se ainda a API Microsoft Speech, a qual é responsável por interpretar a fala captada, possuindo versões para diversos idiomas, tais como Português (Brasil e Portugal), Espanhol e Inglês. A identificação da fala realizada pela API é feita através de uma gramática previamente elaborada em XML (*eXtensible Markup Language*).

Quanto aos *hardwares*, vale destacar a caneta digital I-Pen Mouse, utilizada durante os testes sobre os exercícios. Dentre os dados técnicos desse periférico, possui um tamanho de 142 x 15 x 21 mm, peso de 25 g, com entrada para porta USB e resolução de 800 dpi. No que se refere às principais características de tal dispositivo, este assume a função de um mouse óptico e transforma escritas manuais em texto em formato digital. Vale ressaltar que esta caneta digital não necessita de especificações de *hardware* avançadas, pois requer Pentium CPU ou superiores, sistema operacional Windows a partir da versão Windows98 e MAC OS 9.X ou superiores.

3. HEARING THE WORLD

Em todo o contexto apresentado anteriormente, surge a idéia de elaborar um sistema, intitulado Hearing the World, que auxilie o docente no processo de ensino de pessoas portadoras de deficiência auditiva e surdez, tanto da Língua Portuguesa quanto da Libras. O ambiente educativo tem como intuito prover uma nova forma de ensino, difundir a Língua Brasileira de Sinais, além de combater a exclusão social e, conseqüentemente, a digital. Desta forma, pode-se afirmar que o sistema Hearing the World acompanha os deficientes auditivos e surdos ao logo de suas vidas, iniciando-se na alfabetização e prolongando-se no momento de sua comunicação com ouvintes.

De forma geral, o funcionamento do Hearing the World dá-se a partir do momento em que o educador pronuncia uma letra/palavra em um microfone (acionado por uma tecla do teclado). A partir disto, o som captado é representado, na tela do computador, relacionado à imagem, palavra, símbolo (Libras), sílaba, além de uma animação simulando a pronúncia através dos movimentos labiais, sendo cada item apresentado de acordo com o contexto. Por exemplo, ao ser pronunciada a letra 'A', o sistema apresenta a própria letra, uma palavra que a identifica (como por exemplo, abelha), uma imagem correspondente à palavra, a representação da letra em Libras e, por fim, uma simulação do movimento labial pronunciando tal letra, como ilustrado na Figura 1. Uma vez que sem Libras o aluno tem que se concentrar apenas na leitura de lábios e 50% da mensagem se perde, a representação do movimento labial é um dos fatores essenciais para garantir que o objetivo do sistema seja atingido, enquanto o aluno não possui domínio sobre Libras.

Figura 1: Representação de Fala.

Outro recurso que o sistema disponibiliza é a possibilidade de realizar exercícios como, por exemplo, tracejado, caligrafia, completar e formar palavras, circular e associar imagens às letras/palavras, entre outros. Todos esses exercícios são realizados com o auxílio de uma caneta digital, evitando, desta forma, a utilização do mouse em movimentos mais delicados, que exigem maior coordenação motora. Entretanto, a utilização da caneta digital não restringe o uso do tradicional mouse. Com a utilização da caneta digital, a coordenação motora do aluno será desenvolvida ao longo dos exercícios. A Figura 2 ilustra um dos exercícios disponíveis no sistema, que consiste na associação de imagens a letras na Língua Portuguesa como também em Libras.

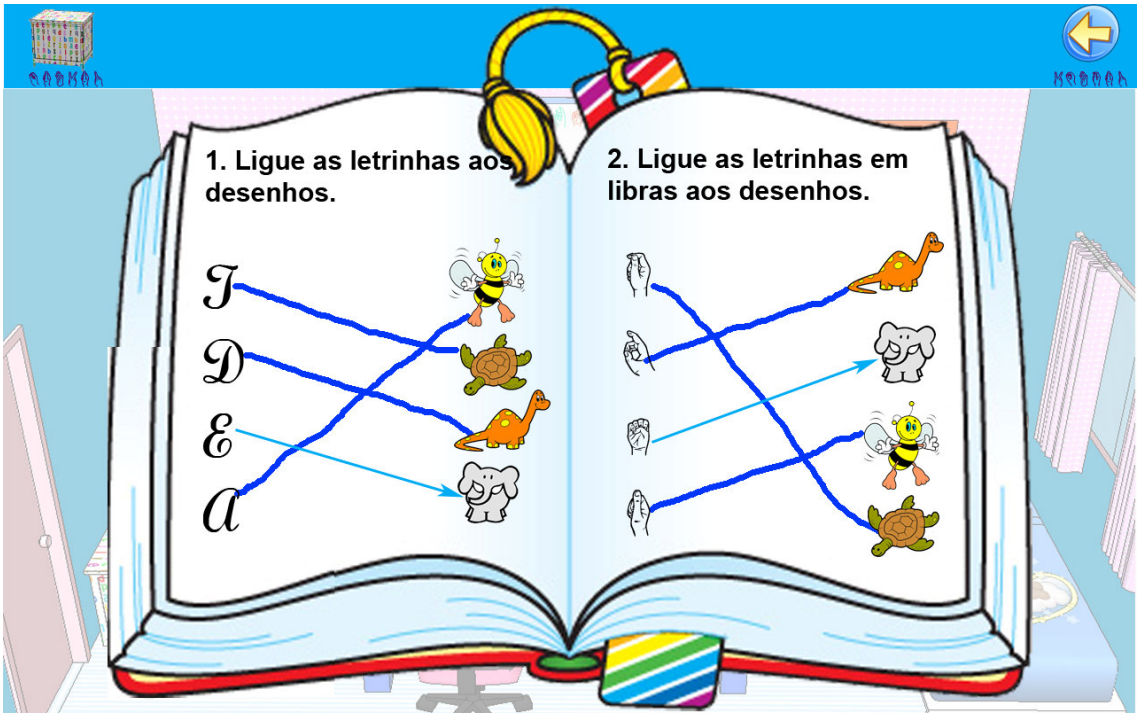


Figura 2: Exemplo de Exercício (ligar).

Vale ressaltar que tanto o conteúdo das aulas ministradas quanto os exercícios realizados pelo aluno podem ser armazenados para estudos posteriores.

Por fim, além de o sistema auxiliar o aluno durante todo o processo de alfabetização disponibiliza, ele também o acompanha ao longo de sua vida, auxiliando-o no processo de interação com a sociedade, através, principalmente, de dispositivos móveis. Por exemplo, enquanto um deficiente auditivo ou surdo estiver utilizando o sistema ao se comunicar com um ouvinte, tudo o que está sendo dito pelo ouvinte aparece para o deficiente auditivo ou surdo, em seu dispositivo, em Libras e em forma de texto na Língua Portuguesa, como demonstrado na Figura 3.

Figura 3: Módulo Comunicação.

É importante enfatizar ainda que o sistema Hearing the World destina-se não somente aos deficientes auditivos e surdos, mas também aos ouvintes. Isso deve-se ao fato do sistema, ao captar a fala, repetir o que fora pronunciado, caso o educador tenha habilitado tal funcionalidade. Além disso, os ouvintes aprenderão Libras, contribuindo para que tal língua seja adotada como objeto de estudo. Os ouvintes ainda desenvolverão sua coordenação motora, ao realizar exercícios com o auxílio da caneta digital. Acentua-se que todos os recursos que o sistema oferece estão adaptados tanto para computadores Desktop quanto para dispositivos móveis como, por exemplo, Ultra Mobile PC, Pocket PC e Laptops.

Ainda, como tentativa de aprimorar o sistema, pretende-se estender o suporte a idiomas de outros países como, por exemplo, o Inglês e o Português de Portugal. Isso, na tentativa de solucionar o problema de comunicação que há entre deficientes auditivos e surdos com ouvintes que atinge não só o Brasil, mas também os demais países.

4. CONSIDERAÇÕES FINAIS

Conforme dados apresentados no início deste documento fica nítido que uma quantidade significativa da população brasileira é portadora de deficiência auditiva e surdez, sendo que, deste total, há uma parcela mínima que possui conhecimento em Libras e encontra-se inserida em um âmbito escolar. E estes poucos que freqüentam alguma instituição de ensino enfrentam situações embaraçosas e preconceituosas, ocasionadas pela falta de preparo e informação por parte dos profissionais da educação, além da falta de apoio, muitas vezes do próprio Governo, embora haja leis que amparam tais pessoas.

Desta forma, acredita-se que a implantação de um sistema tal como o Hearing the World é de grande aceitação, pois auxilia e acompanha deficientes auditivos e surdos desde o seu processo de alfabetização à sua comunicação com ouvintes, a qual atualmente é limitada ao domínio da Língua Brasileira de Sinais. O sistema possibilita também que o aluno desenvolva sua coordenação motora com a utilização de uma caneta digital, para a realização de exercícios. Tal sistema não é limitado ao uso exclusivo de deficientes auditivos e surdos, podendo ser utilizado facilmente por ouvintes, que receberão benefícios semelhantes no que diz respeito à alfabetização. Ainda, possibilita que o usuário não fique restrito apenas em computadores Desktop, fazendo com que, em qualquer lugar, através de um dispositivo móvel, ele possa se comunicar com outras pessoas e, inclusive, utilizar o sistema em outros idiomas.

De forma geral, torna-se evidente que o sistema Hearing the World proporciona aos deficientes auditivos e surdos uma oportunidade de inserção no domínio escolar como também na sociedade como um todo, o qual facilita seu conhecimento e interação. E, além disso, promove a inclusão digital dos mesmos, os quais já apresentam grande facilidade em lidar com esse tipo de tecnologia, hoje de fundamental importância para todas as faixas etárias. Embora haja pessoas que pensam no sistema como algo inviável, no que se diz respeito a recursos financeiros, por este necessitar de ambientes informatizados e outros instrumentos como, por exemplo, caneta digital e dispositivos móveis, constatou-se que tal pensamento é errôneo, uma vez que todas as instituições brasileiras devem possuir laboratórios de informática e o Governo deve apoiar projetos tecnológicos educacionais. Conforme [7], o Governo Federal, através do projeto UCA (Um Computador por Aluno), está se preparando para implantar laptops nas escolas públicas de todos os Estados do país no ano de 2008, visando atingir cerca de 250.000 estudantes. Ainda, através de pesquisas e entrevistas, constatou-se que muitas instituições se mostraram interessadas, afirmando a grande

utilidade do sistema e apoio de empresas, organizações, institutos e, até mesmo, do próprio Governo para sua implantação

REFERÊNCIAS

[1] IBGE. 2000. Disponível em: <http://www.ibge.gov.br>. Acessado em: fevereiro 2007.

[2] 87% das crianças com deficiência auditiva estão fora da escola. Disponível em: <http://www.ciranda.org.br/2004/noticiafull.php?mode=ver&id=52>. Acessado em: fevereiro 2007.

[3] BRASIL. Lei n. 10.436, de 24 de abril de 2002. Dispõe sobre a Língua Brasileira de Sinais – Libras e da outras providências. Diário Oficial da União, Brasília, p.23, 2002.

[4] Legislação da Deficiência Auditiva. **Linguagem de Sinais**, São Paulo, n. 01, p. 13.

[5] MSDN Brasil. **Visual Studio 2005 Professional Edition.** Disponível em: <http://www.msdnbrasil.com.br/visualstudio/Default.aspx>. Acessado em: setembro 2006.

[6] Microsoft Portugal. **O que é o SQL Server 2005?** Disponível em: <http://www.microsoft.com/portugal/sql/prodinfo/overview/what-is-sql-server.mspx>. Acessado em: setembro 2006.

[7] NUBLAT, Johanna; BENEVIDES, Mariana. **Governo Apressa a Expansão de Laptops em Escolas Públicas.** Disponível em: <http://www1.folha.uol.com.br/folha/treinamento/novoemfolha43/te20070629029.shtml>. Acessado em: julho 2007.

Herramientas especiales para plataformas orientadas a la EBW

Mercedes VITTURINI – Perla SEÑAS

[psenas/mvitturi]@cs.uns.edu.ar
Laboratorio de Investigación y Desarrollo en Informática y Educación (LIDInE)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur - Bahía Blanca

Resumen

El avance de la Educación Basada en la Web (EBW) es un hecho que no se discute, la centralización del soporte computacional para el desarrollo de este tipo de formación, en lo que constituye una plataforma de e-learning, también es casi un estándar en estos días. Lo que realmente ocupa hoy a los investigadores y docentes involucrados en esta temática, es cómo diseñar e implementar de la mejor manera las experiencias educativas bajo esta modalidad. Se busca obtener resultados no inferiores a los que se logran con otras modalidades educativas y además solucionar los problemas que aún persisten, tales como algunos aspectos del proceso de evaluación, el desgranamiento de los cursos, el alto costo, entre otros. Un conocimiento claro del tipo de plataformas que existen ayuda no sólo a una elección adecuada entre las disponibles, sino que promueve las investigaciones relacionadas con el diseño de estos entornos, tanto desde la perspectiva de las TICs como desde la educativa. Se presentan en este trabajo dos nuevas herramientas para plataformas de e-learning: el *Hipertexto Referencial*, especialmente creado para asistir al diseñador y organizador de cursos de EBW, y el *Observatorio de Clase Presencial*, como soporte tecnológico de diseños curriculares en modalidad b-learning.

Palabras claves: B-learning - Herramientas Tecnológicas - Hipertexto Referencial - Observatorio de Clase Presencial.

1. Introducción

El e-learning, si se lo define con precisión, se refiere al aprendizaje que se realiza mediado por la tecnología computacional. Puede llevarse a cabo tanto en ambientes cerrados como abiertos, y con diferentes grados de presencialidad: nula, media o total. En el caso de ser total, se refiere a los aprendizajes que se realizan en forma presencial, y la mediación de la tecnología puede darse mediante plataformas aisladas o a través de plataformas distribuidas y conectadas por redes locales o universales. En este último caso se trata de aprendizajes basados en la Web (ABW). Cuando el grado de presencialidad es medio o nulo, parte o todo el proceso se realiza en la modalidad de ABW.

Si bien los modelos de educación no presencial o semi-presencial, son anteriores al e-learning, los estudios actuales se refieren casi en su totalidad a EBW. Las investigaciones en el tema han sufrido una evolución favorable en la que se observa una postura de la tecnología al servicio de las diferentes propuestas educativas. Ello ha desembocado en el estudio de nuevas metodologías educativas y propuestas didácticas aptas para la nueva realidad. Como muy bien expresa Mónica Banzato, "…para poder afrontar el e-learning de manera adecuada, es necesario preocuparse más por las metodologías que por las tecnologías". Por otra parte, el desarrollo de nuevas TICs abre otros horizontes, y los especialistas en Ciencias de la Educación deben conocer la realidad

tecnológica actual, sus posibilidades, limitaciones y tendencias, para explorar, adecuar, y crear nuevas propuestas metodológicas.
Además, siempre se ha buscado que los sistemas de software sean usables, seguros y funcionales. Respecto a esto último, cada vez los requerimientos son mayores. La orientación actual de las investigaciones sobre TICs aptas para aplicaciones educativas basadas en la Web persigue fundamentalmente los siguientes objetivos:

- Desarrollar sistemas de ABW (SABW), siguiendo los lineamientos de la Web Semántica, que permite organizar la información según parámetros de significado, acelerando notablemente los procesos de localización de los objetos didácticos.
- Encontrar esquemas para la representación de conocimiento en Agentes Pedagógicos que sean lo suficientemente flexibles como para ser usados por los expertos de materia, hallar formas precisas para su especificación de tal forma de posibilitar el razonamiento automático y representaciones gráficas adecuadas para su visualización.
- Encontrar recursos adecuados para resolver el tema de la evaluación en el esquema no presencial de la EBW
- Crear nuevas herramientas que amplíen el conjunto estándar disponible en las actuales plataformas de e-learning.

Todos los ítems apuntan al logro de una EBW de calidad. Tienden, entre otros aspectos, a disminuir el valor del cociente horas-educador/número de alumnos, que actualmente es muy alto y a evitar el problema del alto desgranamiento de los cursos, que aún persiste.
En tanto los investigadores de Ciencias de la Educación centran su atención en la búsqueda de nuevas metáforas y paradigmas que se adapten de mejor manera a las nuevas situaciones que se plantean y en el análisis y evaluación de propuestas educativas inscriptas en el modelo e-learning. Los esfuerzos en este sentido también apuntan a la calidad de la enseñanza.
Una plataforma e-learning puede definirse como un entorno computacional de aprendizaje con instrumentos aptos para la gestión de las tareas propias de los actores de una experiencia educativa mediada por tecnología (alumnos, tutores, profesores, autores de los materiales, etc.), de los contenidos didácticos y de la diagramación general de la experiencia. Es el soporte tecnológico que permite llevar a la práctica abstracciones tales como el campus y el aula virtual. Existe gran cantidad de plataformas e-learning y más de una taxonomía sobre ellas. Su conocimiento es importante si se quiere elegir la más adecuada en el contexto del diseño de una experiencia educativa. Las diferentes clasificaciones responden a tópicos fundamentales, tales como: la metáfora subyacente, el grado de generalidad vs. la facilidad de gestión, los perfiles de usuario, el tipo de diseño, el grado de inteligencia, el tipo de comunicación que soporta, su alcance, en relación con la autoría y los derechos de uso y el paradigma de programación en el que se inscribe.

2. Diagramación de experiencias de EBW y herramientas estándar de las plataformas

La tecnología y la comunicación producen profundos cambios en las formas de concebir la educación a distancia. Los nuevos recursos disponibles se basan en la comunicación, la presentación, el acceso y en el uso de la información, bajo la forma de *ambientes educativos virtuales*. Los llamados ambientes virtuales crean espacios con instrumentos tecnológicos que enriquecen la enseñanza a distancia y revalorizan la oportunidad de su uso. Estos espacios se caracterizan por facilitar el intercambio permanente entre profesores y alumnos y entre alumnos y alumnos. Así es posible que datos, proyectos e ideas, mediante los procesos de docencia e investigación, conformen a través de este intercambio una gran comunidad virtual de trabajo

académico colaborativo, generando un punto de encuentro para desarrollar actividades de aprendizajes en grupos. Son muchas más las herramientas y plataformas presentes hoy en la computadora y en la WWW que propician el trabajo en grupos. Sobre el particular precisó que, "la aparición de herramientas que facilitan el trabajo en grupo ha potenciado la realización de procesos de investigación conjunta, proyectos de colaboración e intercambio de opiniones, ideas, experiencias, etc., entre profesores y expertos. Las redes en este sentido ofrecen la tecnología y el vehículo idóneo para realizar acciones de trabajo colaborativo, la distribución de información e intercambio a través de correo electrónico, listas de distribución o servicios de noticia; para acceder a información o cursos específicos dirigidos a la formación y actualización profesional, etc., conformando verdaderas comunidades virtuales" [8].

A continuación se presenta un resumen de las principales herramientas nominadas como colaborativas que generalmente se pueden encontrar hoy en día en los ambientes educativos virtuales:

– Herramientas de comunicación asincrónica: permiten crear grupos de discusión enfocados por un tema, o alrededor de una actividad específica, una meta o proyecto. Entre estas herramientas se destaca el foro. En el diseño de propuestas educativas del tipo EBW es común contar con uno más foros de discusión que propicien el debate, la concertación y el consenso de ideas. De acuerdo a su organización los foros se pueden clasificar en moderados o abiertos, según se prevea o no la existencia de un participante con el rol de moderador. Otras formas de comunicación asincrónica incluyen el uso de correo electrónico y listas de correo.
– Herramientas de comunicación sincrónica: posibilitan la comunicación en tiempo real. Se caracterizan por permitir a todos los participantes acceder a la misma información y al mismo tiempo. La forma más común de comunicación en línea es el chat. El chat permite la interacción con retroalimentación inmediata y abre espacios para crear lazos estrechos entre los miembros. Se pueden plantear espacios de chats abiertos (cualquier miembro se puede unir) o cerrados (requieren de validación de usuarios para el ingreso). Otra forma de comunicación sincrónica es la videoconferencia. La calidad de este tipo de comunicación está condicionada a la calidad del equipo y de la conexión.
– Herramientas para compartir datos y/o documentos: definen una espacialidad común de información. Conforma una base de datos compartida que se puede ofrecer en distintas formas de presentación: pizarras de mensajes, publicación de documentos de estudio, trabajos prácticos, links a sitios de interés, agendas, etc. La incorporación de nuevos datos a la base de información puede ser cerrada, si sólo los miembros con determinado perfil están autorizados a publicar, o abierta a cualquier participante.
– Herramientas de edición colaborativa: proveen una forma de colaboración mediada por la tecnología que permite a un grupo de personas compartir la elaboración de un documento. Tienen capacidades para que múltiples usuarios accedan y modifiquen el mismo documento. Trabajan con control de versiones y formas de protección para evitar inconsistencias.

Cuando se diseña una propuesta de EBW es importante definir el marco pedagógico en el que se inscribirá. Las investigaciones actuales se posicionan fundamentalmente en contextos constructivistas y colaborativos. Al avanzar en el diseño de la experiencia, se llega a un punto donde es necesario seleccionar los recursos que ofrece la plataforma para implementar las estrategias didácticas previamente definidas. Por ejemplo si la táctica es organizar una reunión grupal a partir de la cual se pretende que los alumnos elaboren una conclusión sobre un material dado, teniendo en cuenta la relación contextual de la experiencia en la unidad completa, el tipo de evaluación que realizará (de producción final o de proceso) y algunas de las características de los

aprendices (disponibilidad horaria, equipamiento al que accede, pericia en el manejo de la tecnología, lugar donde accede a la Web, habilidad para argumentar, etc.), el docente seleccionará entonces el recurso de la plataforma que considere más adecuado (chat, lista de correo, foro o grupo). Es importante destacar que el conjunto de recursos estándares disponibles en los entornos más difundidos es aún incompleto como para implementar la diversidad de estrategias didácticas de manera ajustada a las características particulares del grupo de estudiantes. También se investiga sobre herramientas que asistan al docente tanto en la planificación de la propuesta, como en las correcciones de trabajos y visualización rápida del estado del curso, evolución de un grupo o de un alumno particular, etc.

Ulrich Hoppe afirma que los tecnólogos deben preocuparse por inventar o adaptar nuevos instrumentos, y los expertos en educación quienes deben juzgarlos y adoptar los más prometedores. Unos y otros tenemos que pensar más allá de la herramienta única. Las herramientas deben ser ínter operativas y deben apoyar procesos productivos y significativos desde el punto de vista educativo. En este contexto se presentan dos herramientas aptas como para ser incluidas en las plataformas de código abierto más difundidas [4].

3. El Hipertexto Referencial

La tecnología de ninguna manera condiciona el contenido y calidad curricular de una experiencia educativa, sino acompaña su desarrollo y actúa como medio facilitador. Como se mencionó en la sección anterior, los ambientes educativos basados en la Web actuales están provistos de una variedad de herramientas tecnológicas de propósito general. Así, cuando se diseña una propuesta de EBW es importante conocer las ofertas tecnológicas junto con sus posibilidades funcionales y la oportunidad de su inclusión. En este trabajo se presenta el *hipertexto referencial*. Se trata de una nueva herramienta para el diseño y diagramación de cursos a distancia desde lo semántico que colabora con la construcción de Objetos de Aprendizaje (OA) reutilizables, independientes y cohesivos.

3.1. Motivación

Cuando se plantea la elaboración de una propuesta educativa en la forma de asignatura, curso o taller, sea presencial, semi-presencial o a distancia, el primer paso es la definición y organización del contenido curricular del mismo: identificar la necesidad de aprendizaje, definir los objetivos y plantear el material didáctico necesario. En esta definición intervienen directamente los docentes y constituye un basamento fundamental al momento de analizar la calidad final de la propuesta. Tanto para profesores como para estudiantes, el contenido curricular de una asignatura de estudio representa el marco de referencia constante: al docente, porque le ayuda a circunscribir la planificación del curso, así como a estimar horas y recursos; al alumno, como una representación abstracta del curso con las anclas necesarias que le permiten en todo momento visualizar objetivos y conceptos a desarrollar junto con su referencia espacial.

La propuesta de *hipertexto referencial* consiste en un *hipertexto* definido desde el contenido curricular de un curso pensado en la modalidad a distancia. Dicho hipertexto se constituye como una jerarquía que representa la estructura semántica subyacente y a la vez que establece la organización en unidades de estudio y su vinculación explícita con el material didáctico. Así cada curso C_i pensado para una modalidad EBW tiene su propio hipertexto referencial $HR(C_i)$ asociado.

La construcción del $HR(C_i)$ se puede incluir en las primeras etapas del proceso de creación de C_i. De esta manera la exigencia en cuanto a la especificación de los hipervínculos que componen a

HR(C_i) son a su vez un patrón que colabora con la tarea de identificar objetos de aprendizaje funcionales y autocontenidos que conformaran el curso. Esto es, un HR(C_i) funciona como un meta-OA, generador de nuevos OA´s.

3.2. Creación

En el ámbito del aprendizaje basado en tecnología se definen como Objetos de Aprendizaje (OA) a "un recurso digital que permite apoyar el aprendizaje"[2]. Esta definición incluye a una gran variedad de recursos digitales y permite incluir a OA complejos formados a partir de OA más simples. Una característica importante de los OA´s son su potencial de reusabilidad, capacidad generativa y escalabilidad [3].

Aplicando el principio de modularidad, el desarrollo del contenido curricular de un curso C_i se puede distribuir en componentes más pequeños o módulos (M_{ij}). Se espera que varios M_{ij} compongan un C_i. La próxima instancia de descomposición es unidad de estudio (U_{ijk}), donde una U_{ijk} sería equivalente a lo que en un ambiente presencial se asocia con una clase. Un M_{ij} se puede descomponer en uno o más U_{ijk}. Una representación gráfica de la organización de un curso siguiendo esta jerarquía se puede ver en al figura 1.

Paralelamente a la jerarquía de módulos de C_i se define el HR(C_i) como una jerarquía de referencias a contenidos. Cada HR(C_i) queda definido como una jerarquía de hipervínculos restringido por el contenido curricular de un curso y su diagramación en unidades de aprendizaje. Finalmente, a cada U_{ijk} se le asocia el conjunto de material didáctico digital. Así, el desarrollo HR(C_i) es integrado junto con una metodología de desarrollo de los contenidos digitales que conduce al diseño de Objetos de Aprendizaje (OA) independientes, reutilizables y cohesivos. En la figura 1 sobre la izquierda se puede ver el modelo conceptual que identifica a los componentes de un hipertexto referencial.

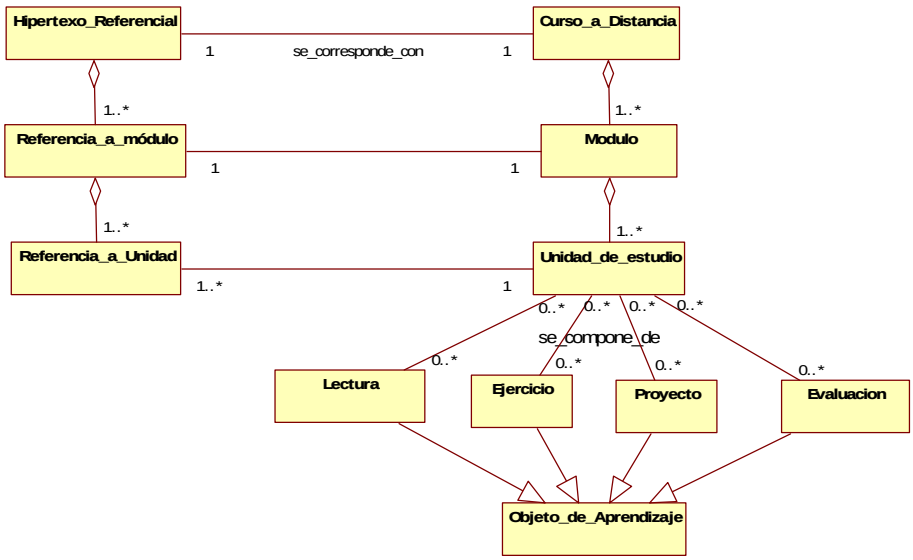


Figura 1 – Relación entre Curso_a_Distancia e Hipertexto_Referencial

Con relación al material didáctico digital, se definió una especialización de OA en objetos de Lectura, Ejercicios y Problemas, Proyectos y Evaluación. Esta especialización de OA es una propuesta y puede extenderse para adecuarse a las necesidades de diferentes disciplinas. En particular se define lo que representa cada uno de estos conceptos como:

– Lecturas: introducen el/los concepto/s a desarrollar en una unidad de estudio. Representan el punto de partida para discutir los temas que se tratan.
– Ejercicios y Problemas: representan la ejercitación práctica para fijar conocimientos asociada a una unidad de estudio. Los ejercicios persiguen el objetivo de aplicar procedimientos conocidos y adquirir soltura en su uso. Los problemas pueden definirse como más amplios y buscan desarrollar en los alumnos habilidades para plantear soluciones. De acuerdo a las necesidades del curso, se pueden juntar o dividir en clases distintas.
– Proyectos: son problemas que requieren de mayor tiempo para trabajarlos, se desarrollan principalmente fuera de una sesión educativa. Pueden requerirse que se resuelvan en forma individual o en grupo.
– Evaluación: son necesarias para tener conocimiento sobre la comprensión y dominio de los temas tratados. Las evaluaciones pueden estar planificadas como evaluaciones para entregar al docente o como autoevaluaciones.

Cada unidad de estudio puede contener, ninguno, uno o más de cada uno de estos elementos; cada uno de ellos constituyendo un OA. Por su parte, la implementación de un OA puede incluir el uso de herramientas tecnológicas. Por ejemplo una unidad de estudio puede requerir de un Proyecto a desarrollar entre varios alumnos usando las facilidades de un editor colaborativo; o un problema puede consistir en crear el mapa conceptual de un texto usando la plataforma MCH. Las distintas herramientas tecnológicas y su relación con OA están fuera del alcance de este trabajo.

3.3. Lectura

Como se describió en la sección anterior, el diseño del HR(C_i) está a cargo del plantel docente de un curso y se confecciona mientras se diagrama del curso. Una vez constituido, se propone poner a disposición esta herramienta como una opción del menú para consultar por cualquier miembro de un curso. Su propiedad hipermedial la transforma en un medio de acceso rápido que permite navegar por el contenido del curso. Su concepción como esquema de contenido, sirve para usar como referente y foco.

4. El Observatorio de Clase Presencial

Es una herramienta tendiente a minimizar el número de encuentros presenciales en cursos con modalidad b-learning, problema no menor en la actualidad cuando hay una tendencia creciente al desarrollo de actividades académicas en forma descentralizada como respuesta a la cada vez más alta probabilidad de no poder coincidir en tiempo y/o espacio.

4.1. Motivación

Se presenta a continuación la justificación de su creación tanto desde lo pedagógico como desde lo tecnológico.

4.1.1. Desde lo pedagógico

La modalidad formativa b-learning (blended learning) es una combinación del sistema presencial y del mediado por TICs. Nació de la necesidad de proponer un sistema formativo que apoye la educación presencial con los beneficios de un sistema e-learning, pero sin asumir los riesgos que supone este último, fundamentalmente en lo referente a su elevado índice de desgranamiento y a las dificultades de aplicación en áreas fundamentalmente prácticas, empíricas o de prueba real. En este sentido “es visto desde diversas teorías pedagógicas como un intento por erigir un sistema que promueve la participación y actitud de implicación activa entre los alumnos, y que mejora y amplía

la comunicación entre los distintos actores formativos más allá del aula fija, lo cual permite concebir una ampliación en tiempo-espacio de las formas de trabajo en grupo colaborativo" [5].

Desde otra perspectiva, la modalidad b-learning se presenta como una forma de educación mixta que parte de la EBW y que se enriquece con los beneficios que aportan los encuentros presenciales para subsanar el problema de la validación de la evaluación final y de los problemas del e-learning antes mencionados. Tal como se muestra en la figura 2, partiendo desde la educación presencial o desde la EBW, la modalidad b-learning se sitúa en un punto intermedio donde el objetivo es recoger los beneficios y disminuir las falencias de ambas posturas extremas. Con una y otra mirada se llega al mismo punto, en ambas habrá encuentros presenciales, entre los que se pueden distinguir:

– Encuentro inicial: donde se establecen los acuerdos y se clarifican los detalles de los instrumentos tecnológicos que se usarán durante todo el curso.
– Encuentros intermedios: estas clases ayudan al docente y a los alumnos a evaluar el progreso del curso, a entablar y estrechar vínculos, que facilitan luego la comunicación no presencial. Todo esto colabora sustancialmente en la disminución del índice de deserción [4].
– Encuentro final en el que los alumnos realizan una prueba de evaluación objetiva de conceptos y/o competencias adquiridas.

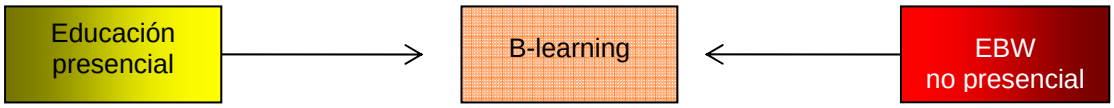


Figura 2 – Modalidad b-learning como educación mixta

Considerando la segunda perspectiva, es importante analizar la situación que se presenta cuando no es posible consensuar horarios y/o lugar físico para los encuentros presenciales. Estos casos aparecen en cursos donde los alumnos viven en poblaciones o hasta en países diferentes, o aún viviendo en el mismo lugar cuando tienen otras obligaciones que les impiden compartir un mismo horario de asistencia a clase. En tales casos aparecen la videoconferencia y la clase grabada como alternativas de reemplazo del encuentro presencial; la primera apta para los casos de desencuentro espacial y la segunda cuando no es posible el acuerdo en lo temporal.

Es así que es posible seguir pensando en una modalidad b-learning aún cuando gran parte de las instancias presenciales, inicial e intermedias, sean reemplazadas por videoconferencias u observaciones de clases grabadas. El reemplazo no es planteado como una mejora sino como una solución alternativa para poder continuar en el paradigma b-learning, aún cuando no estén dadas todas las condiciones. Para que dicho reemplazo sea realmente efectivo, es necesario que tanto la institución educativa como el alumno cuenten con los recursos TICs adecuados. Si bien eso no se puede aplicar hoy a muchos de los casos, es importante que desde el ámbito tecnológico se investigue y se desarrollen las herramientas adecuadas que en un futuro no muy lejano podrán ser aprovechadas por una buena parte de la población.

4.1.2. Desde lo tecnológico

Entre los recursos tecnológicos aptos para reemplazar encuentros presenciales en las propuestas b-learning se pueden citar:

– Laboratorios de acceso remoto: permiten al alumno acceder a través de la Web a laboratorios virtuales o reales. Su aplicación se relaciona directamente con clases empíricas o de prueba real [6][7]. Actualmente es accesible a un alto número de usuarios.
– Videoconferencias: sistemas de comunicación en red en tiempo real que utilizan audio y video. Los modelos recomendados son:
– Multicast, que permite la distribución eficiente del tráfico de la red en forma simultánea a múltiples usuarios y el sistema de videoconferencia VRVS -Virtual Rooms Videoconferencing Syste- o "Sistema de Videoconferencias basado en Salas Virtuales" que sirve para realizar principalmente multiconferencia.
– Tele-inmersión: en este modelo se combina la tecnología de inmersión, los sistemas avanzados de telecomunicación de alta velocidad y tecnologías que permiten reconocer la presencia y el movimiento de individuos, rastrear esta presencia y sus movimientos, para después permitir su proyección en verdaderos entornos de inmersión múltiples, geográficamente distribuidos, en los cuales estos individuos podrían interactuar con modelos generados por la computadora. Esta composición ofrece un nuevo paradigma en la colaboración y comunicación humana. Por ejemplo, cirujanos del otro lado del mundo podrán estar presentes en una sala de operaciones para ofrecer consejo o se puede asistir a una clase a distancia exactamente como si se estuviera presente en el aula. Desde un punto de vista técnico, la clave de la impresión realista que ofrece la teleinmersión es un banco de cámaras digitales hemisférico que captura a los participantes en la sesión desde una gran variedad de ángulos. El mismo sistema hace un seguimiento de los movimientos de la cabeza de los usuarios, los cuales emplean un equipo que incluye gafas polarizadas semejantes a las usadas para ver una película en 3D. La llegada de imágenes ligeramente distintas a cada uno de los ojos permite al cerebro reconstruir la telepresencia tridimensional de una persona que en realidad se encuentra en un estudio muy apartado. La principal dificultad actual es la transmisión de la información y esto se está investigando en el marco del proyecto Internet-2.

El b-learning es considerado hoy como el modelo educativo mediado por TICs de mayor valor, de allí la importancia de poder agregar recursos tecnológicos que palien circunstancias negativas para su aplicación. Los rasgos más significativos de la modalidad son:

– Las TICs son fuente de distribución, almacenamiento y gestión del material docente / estudiante, recuperable, reusable, multi-direccional y multi-formato.
– El método de estudio debe promover la independencia.
– La información aparece descentralizada y distribuida entre todos los agentes del sistema educativo, todo ello soportado en las herramientas multimediales y el soporte Web, posibilitando el desarrollo de las técnicas de enseñanza y aprendizaje más distributivas. Los docentes están en instancias presenciales y en tareas de tutoría más allá del escenario presencial, permitiendo un detallado conocimiento y seguimiento del desarrollo de los alumnos por parte del docente y de los propios estudiantes, presentándose como una puesta abierta a una formación continua y responsable.
– Existe incentivo y adecuación al trabajo colaborativo, ya que al utilizar medios que permiten comunicar y distribuir la información fuera del contexto de la presencialidad, obviando los topes que suponen las variables de tiempo y espacio, la puesta en marcha y consecución de tareas y proyectos en común pueden ser gestionados de forma más cómoda para los discentes, así como abren la posibilidad de la participación conjunta con el docente sin que ello signifique que tenga que existir una coincidencia espacio-temporal de los participantes del proyecto.
– Favorece la evaluación permanente con pruebas cuya estructura de efectividad se adaptan a las respuestas del sujeto. Esto permite mayor precisión y realimentación de las estrategias didácticas y

una visión para el propio alumno que le ayuda a analizar su propio rendimiento y a orientarse en la elección de los métodos de aprendizaje.

4.2. Creación

La creación de la herramienta comenzó una vez que se definió su utilidad, alcance y funcionalidad. Se realizó el modelado y se precisaron las formas de visualización, antes de concretar la implementación. La herramienta construida puede insertarse luego en la plataforma e-learning como un recurso más. Con ella el alumno podrá observar una clase presencial focalizando la entidad que más le interesa en los diferentes momentos. Es importante destacar en este punto, cuáles son esas posibles entidades. En una clase presencial tradicional coexisten personas que cumplen diferentes roles: docentes y alumnos que asumen roles más o menos activos, otros objetos educativos como: contenidos, recursos didácticos, materiales, canales de comunicación. Todos estos elementos, entre otros, constituyen una clase presencial, proveyendo foco y contexto apropiados que dan lugar a un entorno común de aprendizaje.

4.2.1.Modelo

Al preguntarse por el verdadero valor de la instancia presencial para el alumno, puede advertirse que va más allá de lo que le brinda el profesor. En este modelo se rescatan cuatro entidades fundamentales que pueden implementarse sin necesidad de recurrir a tecnologías para videoconferencias o tele inmersión, ya que el propósito es contar con un recurso de acceso asincrónico con más funcionalidades que un video tradicional:

El *profesor*: muestra al docente en acción, explicando, haciendo u orientando alguna experiencia, interactuando con los alumnos.
Los *compañeros*: da una vista panorámica del alumnado presente, tal cual lo puede observar el profesor. Puede advertirse la expresión de sus rostros y su actitud general, elementos que colaboran a la retroalimentación del que los está observando.
El *foco de desarrollo*: se refiere al lugar donde el docente plasma sus explicaciones. Puede ser una pizarra, una pantalla de proyección, una mesada de laboratorio, un banco de prueba, etc.
El *diálogo en la clase*: muestra un registro de los diálogos que se producen a lo largo de la clase (preguntas y respuestas). Permite un reconocimiento de las interacciones dadas entre los presentes.

4.2.2. Visualización

A partir de la identificación de las cuatro entidades principales: el profesor, el alumnado, el foco de desarrollo y el diálogo en la clase, se diseño el modelo de la herramienta Observatorio Presencial tal como se puede observar en la figura 3. Se trata de proveer la filmación de una clase presencial en tres ángulos más un reproductor de texto, todo ello visualizable en cuatro paneles ajustables sincronizados, cada uno con la vista de una de las entidades. La definición del espacio ocupado por cada uno de los paneles la establece el observador que puede ajustarlos según su foco de su interés. La herramienta se completa con la barra de comandos central que provee las funcionalidades básicas para la reproducción sincrónica de las cuatro vistas.

Las cuatro entidades mencionadas constituyen así la abstracción de la clase presencial que le mostrará el observatorio al alumno que por diferentes razones no ha podido participar de ella.
El observatorio dispone de un modo autor en el que se ofrece un cuadro de diálogo al encargado de componer la clase a ser visualizada, para que pueda ingresar las referencias necesarias para armar el producto.

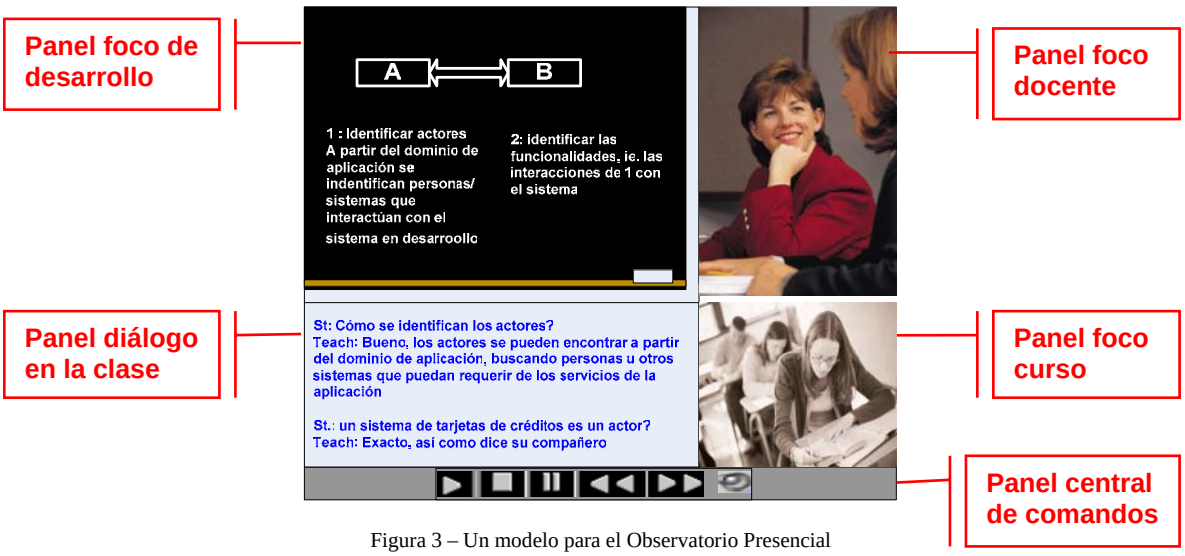


Figura 3 – Un modelo para el Observatorio Presencial

4.3. Organización de una experiencia usando el observatorio

Cuando los alumnos se inscriben, deberán manifestar obligatoriamente los días y horarios en los que podrán participar de encuentros presenciales. Con esta información se comienza con la diagramación de los encuentros. Deben poder consensuarse horarios de clases de tal forma que en cada una de ellas pueda haber un grupo de alumnos presentes que no necesariamente será siempre el mismo. De esta manera, podrán realizarse los encuentros, que serán filmados, cumpliéndose así con dos objetivos: por un lado se concreta la clase presencial para los alumnos que asisten, y por otro se obtienen los videos que compondrán luego el Observatorio para los alumnos que no han podido concurrir.

Es importante que el docente al desarrollar la clase, tenga presente a todos sus alumnos, no sólo a los que están en la clase sino también a los que la observarán en tiempo diferido, por ejemplo cuando hace referencias a correcciones de trabajos, hace recomendaciones o sugerencias, etc.

5. Conclusiones

Se han presentado en este trabajo dos nuevas herramientas para plataformas de e-learning, una relacionada con la organización y acceso a cursos desde lo conceptual y otra como soporte tecnológico de diseños curriculares en modalidad b-learning, ambas con un fuerte potencial tanto para el quehacer docente como para el aprovechamiento de los alumnos en la construcción de conocimiento.

6. Referencias

[1] Galvis Panqueva, A. et. all. “Ambientes Educativos para la Era de la Informática”. Santa Fe de Bogotá, Mayo del 2001.

[2] Wiley, D. “Connecting Learnig Objects to instructional design theory: a definition, a metaphor, and taxonomy”. Utah State University. 2002.

[3] Wiley, D. “The instructional Use on Learning Objects”. URL http://www.reusability.org/read/ 2000.

[4] Ulrich Hoppe. B-learning. e-Learning Papers.www.elearningpapers.eu. Enero 2007

[5] Brennan, M. Blended Learning and Business Change. Chief Learning Officer Magazine. Enero 2004. http://www.clomedia.com

[6] Del Castillo R. et. all. Experiencia con Laboratorio Remoto Colaborativo. TE&ET. 2006

[7] De Giusti, A., Naiouf, M. y Pousa A.Laboratorio de Procesamiento Paralelo Multi-Cluster accesible vía WEB. . TE&ET. 2007.

[8] De Benito, B. (2000). *Posibilidades Educativas de las "WEBTOOLS"*. Palma, España. Ediciones de la Universitat de les Illes Balears.

[9] Derntl, M., Mangler, J. "Web Services for Blended Learning Patterns," IEEE International Conference on Advanced Learning Technologies (ICALT'04), 2004

[10] González Videgaray, M. Assessment of Student and Teacher's Reaction in a Blended Learning Model for Higher Educatio. http://www.uv.es/RELIEVE/v13n1/RELIEVEv13n1

[11] Andrej, K. and Bieliková, M. Improving adaptation in web-based educational hypermedia by means of knowledge discovery. Proceedings of the sixteenth ACM conference on Hypertext and hypermedia. 2005.

[12] ATutor. Información disponible en el sitio oficial (http://www.atutor.ca)

[13] Banzatto, M., Corcione D. "La tutoría en red-Funciones del Tutor on-line". Tutor on-line. Universidad de Salamanca. 2006.

Identifier Management and Resolution: conforming the IEEE Standard for Learning Object Metadata

Roberto J. Godoy
Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral
Santa Fe, 3000, Argentina
rjgodoy@fich.unl.edu.ar

Hugo Minni
Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral
Santa Fe, 3000, Argentina
hminni@fich.unl.edu.ar

Abstract

Uniform Resource Identifiers are an integral part of the current Architecture of the World Wide Web, as well as the Semantic Web initiative.

This work analyzes the implications and possibilities of using Universal Resource Names as unique and persistent identifiers in systems for management of decentralized content and federated collections. Particularly, discussion focuses on applying such identifiers on the context of a learning object repository that the authors are developing at Universidad Nacional del Litoral, according to the IEEE 1484.12.1 standard for Learning Object Metadata.

It is explained why Uniform Resource Locators are inadequate, and why Universal Resource Names are preferable. A standardized resolution service over Hypertext Transfer Protocol is recommended for locating resources, and usage of Uniform Resource Characteristics for accessing Learning Object Metadata is proposed. Finally, content-negotiation mechanisms, for selecting the best representation among several format or language variants, are outlined.

The proposed naming schema provides a double-indirection mechanism, comparable to the Human-Friendly Names approach proposed by Ballintijn, van Steen, and Tanenbaum for improving scalability and usability in naming replicated resources.

Keywords: Web Systems, Service-oriented Architecture, Learning Objects, Education Informatics

1 INTRODUCTION

In the last years, there has been an ongoing discussion about Uniform Resource Identifiers (URIs) and their advantages in comparison with Uniform Resource Locators (URLs) [1],[2]. URIs are an integral part of the current Architecture of the World Wide Web, as well as the Semantic Web initiative [3].

> "Global naming leads to global network effects (...) To benefit from and increase the value of the World Wide Web, (...) a resource should have an associated URI if another party might reasonably want to create a hypertext link to it, make or refute assertions about it, retrieve or cache a representation of it, include all or part of it by reference into another representation, annotate it, or perform other operations on it. Software developers should expect that sharing URIs across applications will be useful, even if that utility is not initially evident." [4]

The election of *unique* and *persistent* identifiers is an important matter when dealing with decentralized content management and federated collections, which are often loose constructs without significant central authority [5]. Additionally, implementing *standardized resolution methods* is indispensable for large-scale deployment and interoperability with other systems.

In a previous work, the authors have collected and analyzed several protocols concerning these identifiers [6]. Their interest is to utilize URIs as identifiers on a Knowledge Repository they are developing, which will be used in a university educational context [7].

It must be noted that although the former analysis takes place within the specific scope of Learning Object Metadata (LOM), some results may be applied to general applications that make use of URL.

2 BACKGROUND

The data entities in these systems are denominated *learning objects* (LO). They may be digital or non-digital and may be used for learning, education or training [8]. Metadata is required in order to describe LO, enabling learners and instructors to search, evaluate and utilize them; and standards compliance leads to a uniform style, enhancing the possibilities of sharing, reuse, and exchange of contents. The IEEE standard for *Learning Object Metadata* (LOM) was chosen among several others because it specifies a conceptual data schema (the "*base schema*") that emphasizes on the minimal set of attributes needed to allow these LO to be managed and located.

2.1 Naming requirements

Each LO and each metadata instance in the base schema is identified by a pair composed by two elements: a `Catalog`, which is the name of an identification or cataloging scheme, and an `Entry`, which is the value of the identifier itself and belongs to the given catalog. For instance, URIs may be used as identifier entries under the "`URI`" catalog; other possible catalogs include `ISBN`, `LCCN`, `ARIADNE`, etc.

Identifiers must be *unique* in the sense they univocally identify a resource, albeit a single resource may be identified by more than one identifier.

2.2 LOM identifiers and URI

> "A URI can be further classified as a locator, a name, or both. The term URL refers to the subset of URIs that, in addition to identifying a resource, provide a means of locating the resource by describing its primary access mechanism (e.g., its network "location"). The term "Universal Resource Name" has been used historically to refer to both URI under the `urn` scheme [9], which are required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable (...)" [10]

Among URI schemes and Universal Resource Name (URN) namespaces, `urn:fdc` [5] was found to best fulfill the requirements (though others schemes or namespaces may be used in particular cases). On the other hand, URLs are not suitable as identifiers, because they are inherently non-persistent. [6]

2.3 URN resolution

URN resolution is the process of translating a URN into *Uniform Resource Locator* (URL) or *Uniform Resource Characteristics* (URC) [11]. Resolution services, defined in RFC 2483 [12], provide a uniform interface for performing these conversions. They are given mnemonic names, such as N2L (which stands for URN to URL), N2R (URN to resource), etc. Some services yield a single result, while others yield multiple results (e.g., *all* the locations of a resource). There are also services that carry out the inverse conversion (e.g., they gather the URNs for a given URL).

THTTP (Trivial Convention for using HTTP in URN Resolution) protocol [13] specifies how to access resolution services via traditional Hypertext Transfer Protocol (HTTP) GET requests. The services implemented by THTTP are shown in Fig. 1.

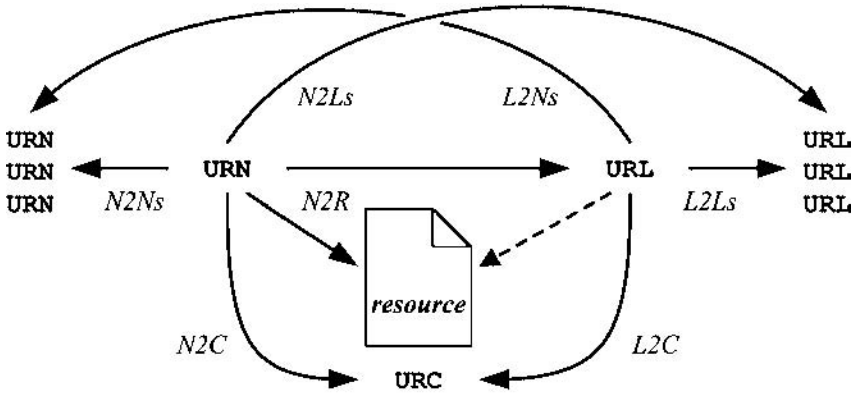


Figure 1: Resolution services of THTTP protocol.

3 PROPOSAL

3.1 Use of URN as resource identifiers

There are two obvious advantages of using URLs as if they were identifiers: resources that are accessible via HTTP or File Transfer Protocol (FTP) already have a URI of the URL kind, and it is straightforward to get the resource from its identifier.

Despite of these apparent advantages, the intended semantics of URL is to locate, not to identify. Identifiers must be independent from the resource location and it must be possible to keep the same identifier after moving the resource. Additionally, a LO may be tagged as "unavailable", or it may be of a non-digital nature (i.e. a physical resource whose metadata is recorded in the system); in this situation it *cannot* be associated with a true URL which dereferences it.

Despite they are less common than URL, and despite of their need of namespace management, URN identifiers are adequate for addressing these problems. Anyway, if persistence is honored, it follows that URL-based identifiers will become outdated; and supporting deprecated or fake URLs requires as much effort as supporting identifiers that do not disclose the location.

3.2 Access LOM metadata as URC

Uniform Resource Characteristics (URC) are generic metadata about resources. They are vaguely defined in RFC 2483 as descriptions that may include "a bibliographic citation, a digital signature, or a revision history", but the content of any response to a URC request is not specified [12]. Since LO are *described* by metadata instances, it seems natural to access LOM metadata as Uniform Resource Characteristics (URC) via THTTP services N2C/L2C.

This approach provides a uniform interface for accessing LOM instances, which is similar to the resolution methods for accessing resources (N2R) or locations (N2L), thus avoiding application-specific retrieval mechanisms.

The type of URC to be returned is specified by a Multipurpose Internet Mail Extensions (MIME) [14] type, which does not only identifies the format of the result (as usual), but also its content. This requires a semantically unambiguous MIME type in order to indicate that LOM XML (Extensible Markup Language) metadata is requested, instead of other metadata (which may be optionally supported).

The MIME type `text/xml` is too general because it does not state that LOM is specifically required. A hypothetical `text/lom` type (which does not exist) would not be correct because LOM may be also encoded as Resource Description Framework (RDF) and other bindings may be defined in the future.

The `+xml` suffix [15] was defined for dealing with XML-based MIME types. For instance, some applications would be able to understand entities of `text/lom+xml` type, while others (e.g. an XML viewer) will treat them as generic XML documents. Moreover, applications without explicit support for `text/xml` will treat them as plain text.

In this case, `text/x.lom+xml` should be used because `text/lom+xml` does not exist. The `x.` prefix implies the subtype belongs to the unregistered experimental tree. If this approach proves to be useful, a registration proposal may be submitted to the Internet Assigned Numbers Authority (IANA). (As a side note, the LOM RDF encoding cannot be expressed in the same way, because there is no `+rdf` suffix.)

3.3 Content negotiation mechanisms

Resources may be available in multiple languages and formats (e.g., slides as both application/vnd.ms-powerpoint and application/pdf). The mechanism for selecting the appropriate representation when servicing a request is known as content negotiation [16].

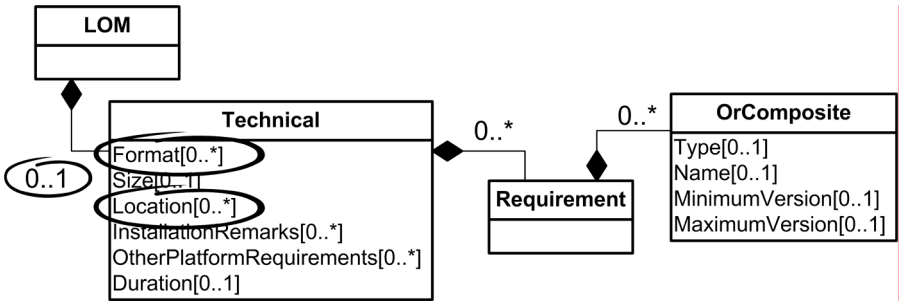


Figure 2: `Technical` element from LOM base schema.

One obstacle with the current version of the LOM base schema is that it only *lists* different representations of a resource (variants), but there is no way for distinguishing which variant corresponds with each location (Fig. 2).

Agent-driven negotiation —where the user would be able to select a variant based on other attributes (such as technical requirements) (Fig. 3(a))— is not possible because the information provided by LOM schema is incomplete.

Nevertheless, the appropriate resource may be determined on the server side by using THTTP for supplying the missing metadata. Since THTTP relies on negotiation mechanisms from HTTP protocol, this strategy allows resolution services to be accessed by general-purpose user agents (e.g. web browsers) and provides *server-driven negotiation* (Fig. 3(b)) by means of standard HTTP message-headers `Accept` and `Accept-Language`.

As a drawback, this strategy requires additional data structures for keeping the descriptors (e.g. language, format, and location) of each variant, which are not separately stored by LOM.

The IEEE Learning Technology Standards Committee is currently working on a new version of the standard [17]. As a result, the multiplicity of `Technical` element might be modified and this agent-driven negotiation would be possible. This modification would also allow THTTP to be implemented over LOM, without additional structures.

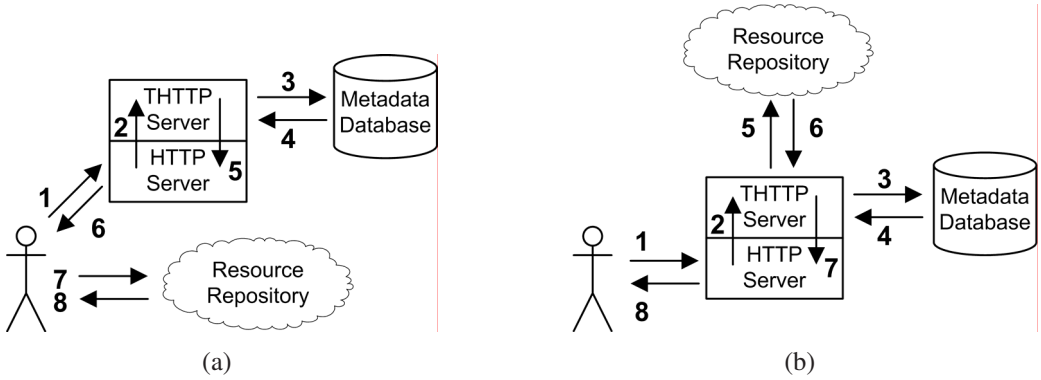


Figure 3: (a) agent-driven negotiation and (b) server-driven negotiation.

3.4 URN for vCard externalization

Personal information about authors, editors, content providers, and other actors who contribute to the LO lifecycle, is represented in LOM as vCard 3.0 [18] entities, which are embedded into each

metadata instance, as shown in Fig. 4(a). The authors have recommended a LOM-compliant externalization strategy (Fig.4(b)) for storing that information in a normal form: metadata instances should contain a minimal vCard representation, and refer external vCard resources where additional (or updated) information would be located [7].

These references, indicated by means of the `source` attribute within the embedded vCard, are themselves URIs. In the original proposal `ldap:` (a URL schema) was suggested, as it was also exemplified in RFC 2425 [19]. However, since the `source` attribute accepts any kind of URI, persistent identifiers (i.e. URNs) may be specified. They may be subject of the resolution mechanism explained in previous sections without introducing additional complexity to the system.

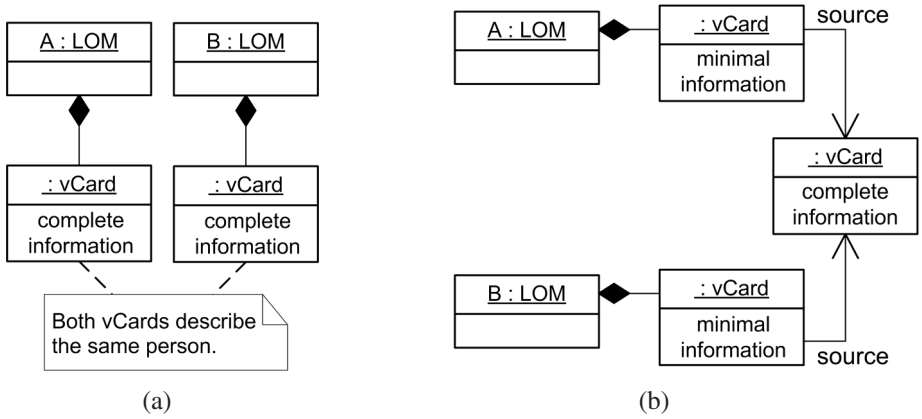


Figure 4: (a) embedded vCard entities and (b) references to external vCard.

3.5 URN as a high-level indirection layer

The LOM base schema provides an specific element (`Technical.Location`) for specifying how the contents may be accessed. This element accepts a URI as value, but this URI is intended to resolve to the content location, and not to identify the LO itself as the LOM identifiers do.

A two-step resolution process may be implemented, which is similar to the *Human-Friendly Names* (HFN) approach by Ballintijn, van Steen, and Tanenbaum [20], shown in Fig. 5(a). They proposed a second indirection layer, in addition to URN/URL mechanism, in order to identify resources with "names that are easy to share and remember", while URN were regarded as machine-oriented identifiers for grouping several replicas[1].

The resolution method proposed in this paper allows this kind of two-layer resolution within the scope of LOM standard: LOs are assigned with high-level human-oriented URNs, and the location of their contents is specified by other low-level URNs, as shown in Fig. 5(b). In turn, each low-level URN resolves to one or more URLs, which are either *mirrors* (i.e. alternate locations) or *variants* of the resource.

[1]They introduced Human-Friendly Names (HFN) as a URI scheme instead of a URN namespace. As a historical note, there was no human-oriented general purpose URNs namespaces by the time they wrote their article, but this situation has changed since then.

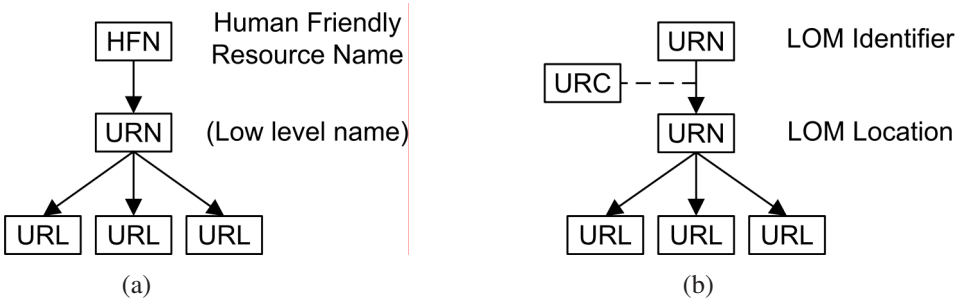


Figure 5: (a) Naming scheme using HFN combined with a URN, as proposed in [20], and (b) two-step URN-based resolution process.

4 CONCLUSION

As stated by Duncan, a learning object repository is a complicated system because it must deal with granularity, versions, relations between entities, and relations between metadata and entities [21]. The complexity increases under the requirement of supporting federated collections of decentralized content.

Although there is a strong theoretical background about URN identifiers, it was found that common URL schemes are normally used, and Learning Object implementations does not take full advantage of difference between names and identifiers. (For instance, Powell et al. explicitly recommend the `http:` scheme [2],[22])

This work shows the advantages of URN in comparison with URL. URNs are preferable because they have identifier semantics and they are intrinsically persistent. In addition, several benefits from its adoption are explained.

THTTP protocol is suggested for implementing resolution services, because of three reasons:

- its implementation is very simple,
- its specification underwent enough revision as per RFC procedures,
- web browsers and other HTTP user agents are already enabled to access resources with no need for specialized software.

A method for encoding metadata requests by means of THTTP services is proposed, and data retrieval is enhanced with server-driven negotiation of contents. The resolution scheme is not restricted to LO; indeed, it extends to other resources such as vCards, allowing references to personal information to be normalized according to IEEE LOM standard. This is a very important feature for the design of the repository at Universidad Nacional del Litoral, in which not only LO but also contributors are considered first class entities.

ACRONYMS

FTP File Transfer Protocol

HFN Human-Friendly Names [20]

HTTP Hypertext Transfer Protocol [16]

IANA	Internet Assigned Numbers Authority
L2C	URL to URC a THTTP resolution service [13]
LO	Learning Object [8]
LOM	Learning Object Metadata [8]
N2L	URN to URL (a THTTP resolution service) [13]
N2R	URN to resource (a THTTP resolution service) [13]
N2C	URN to URC (a THTTP resolution service) [13]
URC	Uniform Resource Characteristics [11]
URI	Uniform Resource Identifier [10]
URN	Universal Resource Name (a URI scheme) [9]
URL	Uniform Resource Locator (a subset of URI)
MIME	Multipurpose Internet Mail Extensions [14]
THTTP	Trivial Convention for using HTTP in URN Resolution [13]
RDF	Resource Description Framework
XML	Extensible Markup Language

REFERENCES

[1] D. Booth, "URIs and the myth of resource identity," *Identity, Reference, and the Web, WWW2006 Workshop*, 2006.

[2] A. Powell, P. Johnston, L. Campbell, and P. Barker, "Guidelines for using resource identifiers in Dublin Core metadata and IEEE LOM," Dublin Core Metadata Initiative, DCMI Recommended Resource, Apr. 2005, http://www.ukoln.ac.uk/metadata/dcmi-ieee/identifiers/ (last access 2007-05).

[3] N. H. Shadbolt and T. W. Berners-Lee, "The semantic web revisited," *Intelligent Systems, IEEE*, vol. 21, no. 3, pp. 96 – 101, Feb. 2006.

[4] I. Jacobs, N. Walsh *et al.*, "Architecture of the world wide web, volume one," World Wide Web Consortium, W3C Recommendation REC-webarch-20041215, Dec. 2004.

[5] D. Tessman, "A uniform resource name (URN) namespace for federated content," Internet Engineering Task Force, RFC 4198, Nov. 2005.

[6] R. J. Godoy and H. Minni, "Asignación y resolución de identificadores para un repositorio de objetos de aprendizaje basado en LOM," in *IX Workshop de Investigadores en Ciencias de la Computación*, Trelew, 2007, pp. pp658–662.

[7] R. J. Godoy, H. Minni, G. Zarza, and H. Loyarte, "Design criteria for the development of an institutional learning object repository," in *Proceedings of XII Argentine Congress on Computer Science*, San Luis, 2006.

[8] Learning Technology Standards Committee, "IEEE standard for learning object metadata," Institute of Electrical and Electronics Engineers, New York, IEEE Standard 1484.12.1, 2002.

[9] R. Moats, "URN syntax," Internet Engineering Task Force, RFC 2141, May 1997.

[10] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform resource identifier (URI): generic syntax," Internet Engineering Task Force, RFC 3986, Jan. 2005.

[11] K. Sollins, "Architectural principles of uniform resource name resolution," Internet Engineering Task Force, RFC 2276, Jan. 1998.

[12] M. Mealling and R. Daniel, "URI resolution services necessary for URN resolution," Internet Engineering Task Force, RFC 2483, Jan. 1999.

[13] R. Daniel, "A trivial convention for using HTTP in URN resolution," Internet Engineering Task Force, RFC 2169, Jun. 1997.

[14] N. Freed and N. Borenstein, "Multipurpose internet mail extensions (MIME) part two: Media types," Internet Engineering Task Force, RFC 2046, Nov. 1996.

[15] M. Murata, S. Laurent, and D. Kohn, "XML media types," Internet Engineering Task Force, RFC 3023, Jan. 2001.

[16] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol – HTTP/1.1," Internet Engineering Task Force, RFC 2616, Jun. 1999.

[17] W. Hodgins, E. Duval *et al.*, "LOM corrigenda review committe," Learning Technology Standards Committee, Draft Minutes, 2007.

[18] F. Dawson and T. Howes, "vCard MIME directory profile," Internet Engineering Task Force, RFC 2426, Sep. 1998.

[19] T. Howes, M. Smith, and F. Dawson, "A MIME Content-Type for directory information," Internet Engineering Task Force, RFC 2425, Sep. 1998.

[20] G. Ballintijn, M. van Steen, and A. Tanenbaum, "Scalable human-friendly resource names," *Internet Computing, IEEE*, vol. 5, no. 5, pp. 20–27, Oct. 2001.

[21] C. Duncan, "Use cases for persistent identifiers," in *DCC Workshop on Persistent Identifiers*, Glasgow, Jul. 2005.

[22] A. Powell, "Identifiers - requirements and issues," in *DCC Workshop on Persistent Identifiers*, Glasgow, Jul. 2005.

Incorporando Diagramas $\mathcal{M}$OISE$^{+}_{EDU}$ para Reorganização de um Curso Técnico em Edificações

Glaucius Décio Duarte[1]
Curso Técnico em Edificações, CEFET-RS
Pelotas/RS, 96015-360, Brasil
glaucius@cefetrs.tche.br

Antônio Carlos da Rocha Costa
ESIN, UCPel
Pelotas/RS, 96010-000, Brasil
rocha@ucpel.tche.br

e

Mara Lúcia Fernandes Carneiro
Instituto de Psicologia, UFRGS
Porto Alegre/RS, 90035-003, Brasil
mara.carneiro@ufrgs.br

Abstract

This work presents a visual methodology, with use of a model based on the theory of multiagent systems for the reorganization of a technician course in constructions. A new convention is presented to be incorporated model $\mathcal{M}$OISE^{+}, developed for the organization of an multiagent, applicable system to the educational case. This model, defined with $\mathcal{M}$OISE$^{+}_{EDU}$, passes to be applied for the reorganization of the structural, functional and deontics specifications of the educational organizations, with the objective to facilitate the management of courses that are submitted to the curricular or teachers alterations.

Keywords: Educational management, organization and reorganization of courses, multiagent systems.

Resumo

Este trabalho apresenta uma metodologia visual, com utilização de um modelo fundamentado na teoria de sistemas multiagentes para a reorganização de um curso técnico em edificações. Apresenta-se uma nova convenção a ser incorporada ao modelo $\mathcal{M}$OISE^{+}, desenvolvido para a organização de um sistema multiagente, aplicável ao caso educacional. Este modelo, definido com $\mathcal{M}$OISE$^{+}_{EDU}$, passa a ser aplicado para a reorganização das especificações estruturais, funcionais e deônticas das organizações educacionais, com o objetivo de facilitar a gestão de cursos que estejam submetidos à alterações curriculares ou do corpo docente.

Palavras-chave: Gestão educacional, Organização e Reorganização de Cursos, Sistemas Multiagentes.

1 Doutorando no Programa de Pós-Graduação em Informática na Educação (PGIE/UFRGS).

1 INTRODUÇÃO

A educação profissional de nível técnico no Brasil, regulamentada pelo Ministério da Educação (MEC), inclui a necessidade de atualização permanente dos cursos e currículos, estabelecida no item VI, artigo 3.º, da resolução nº 4/99 do Conselho Nacional de Educação (CNE) e Câmara de Educação Básica (CEB). Este trabalho propõe uma metodologia para instrumentalização dessa necessidade, utilizando diagramas especialmente desenvolvidos para a modelagem de cursos nesta modalidade de ensino. Estes podem ser considerados como organizações educacionais, que estão constantemente passando por processos de reorganização.

A reorganização de um curso técnico, a cada semestre letivo, exige do professor comprometido com o papel de coordenador de curso, a árdua tarefa de reformulação dos horários de unidades curriculares, envolvendo a inclusão, manutenção ou exclusão de comprometimentos de professores com as turmas.

Este trabalho, considera a existência de um processo permanente de reconstrução curricular e docente, agravado pelo alto índice de professores substitutos, que de acordo com a legislação vigente, podem permanecer somente pelo prazo máximo de dois anos.

1.1 Curso Técnico em Edificações

O Curso Técnico em Edificações (EDI) ofertado pelo CEFET-RS, de acordo com Sastre et al. [8], tem sua existência justificada na medida em que visa fundamentalmente à formação de profissionais para atender a demanda da indústria da Construção Civil, setor responsável pelo suprimento do déficit habitacional no Brasil. A inexistência de um plano habitacional, de forma a eliminar as carências de unidades residenciais, estimadas em mais de vinte milhões, faz com que este déficit seja progressivo a cada ano.

As empresas construtoras, conscientes de que o alcance social vai muito além do retorno financeiro e da sua própria sobrevivência, resolveram, desde a década de oitenta, investir no setor recursos próprios através de planos de financiamento direto aos usuários. Essas empresas estão cientes de que a construção civil é um grande pólo gerador de empregos para trabalhadores de diversos níveis de formação, desde o operário mais simples (servente) até o profissional de nível superior (engenheiro e arquiteto) contando estes, com o apoio imprescindível do técnico de nível médio.

Assim, o objetivo principal do curso consiste na formação de um profissional legalmente habilitado, que possua competências para atuar em: escritórios de projetos, orçamentos, levantamentos de material para estimativa de custo, laboratórios de materiais de construção e solos, levantamentos topográficos, planejamento e execução de obras de construção civil, coordenação de equipes de trabalho, seleção e treinamento de pessoal, realização de interfaces entre áreas técnicas e administrativa das construtoras.

1.2 Reformas Curriculares e o Curso Técnico em Edificações

Toda reforma pressupõe uma intenção explícita de mudar, de melhorar, corrigir, aperfeiçoar, acrescentar, suprimir e indicar, se necessário, uma provável ruptura com o modelo existente, desvinculando-o do mesmo para criar uma nova proposta de trabalho, com objetivos que atendam às reais necessidades do mercado e das novas formas de relações de capital e trabalho. O surgimento de novas tecnologias, que evoluem a cada ano, faz com que velhos conceitos e formas de atuar sejam repensados.

Adaptar EDI às novas exigências do mercado, aliado à legislação vigente, consiste no objetivo maior das últimas reformas implantadas no CEFET-RS. As reformas não ocorrem somente em nível curricular, mas são fundamentalmente estruturais, tendo como uma das referências o sistema educacional estabelecido pela legislação educativa, assim como pelos exemplos e experiências

adquiridas desde a implantação do curso na instituição.

Considerou-se, ainda, que os docentes comprometidos com o curso devem ter sua atuação fundamentada no desenvolvimento e aprofundamento de conhecimentos e habilidades em conteúdos específicos em suas áreas de atuação.

Para embasar a construção da metodologia proposta a seguir, optou-se pela realização de um experimento inicial cujo andamento pudesse ir fornecendo o *feedback* necessário para a consolidação de tal construção.

2 DESENVOLVIMENTO

2.1 Organização de EDI

A organização do curso (EDI) foi inicialmente modelada, em função das seguintes necessidades:

- Estruturação do curso em módulos semestrais;
- Distribuição das disciplinas nos módulos semestrais;
- Comprometimento de docentes com os módulos;
- Comprometimento de docentes com as áreas do curso;
- Comprometimento de docentes com as disciplinas;
- Comprometimento de docentes com os conteúdos a serem ministrados nas disciplinas;
- Estruturação do espaço físico das instalações do curso;
- Alocação de espaços para equipamentos existentes no curso;
- Especificações didáticas;
- Especificações pedagógicas;
- Comprometimento de docentes com tarefas destinadas à organização do curso;
- Comprometimento do curso e docentes com projetos de extensão ou pesquisa.

2.2 Modelo $\mathcal{M}$OISE$^+$

O modelo organizacional $\mathcal{M}$OISE$^+$ [5], estabelecido com base na tecnologia de sistemas multiagentes (SMA), permite estabelecer precisamente quais os componentes que formam uma organização e como estes podem contribuir para a finalidade do SMA [1][9]. Os diagramas apresentados para este modelo, lembram de forma significativa, as convenções definidas pela *UML* [6].

Além desta função ontológica, o modelo $\mathcal{M}$OISE$^+$ foi desenvolvido para auxiliar, em uma etapa posterior, um processo de reorganização. Apresenta, portanto, características que suportam tanto a análise quanto o projeto de novas organizações.

Ao analisar este modelo, verifica-se que a Especificação Organizacional (EO) é estática, sendo que a criação de entidades segue a especificação definida pelas Entidades da Organização (EnO). De forma análoga, há duas noções de grupo: a especificação de um tipo de grupo e um grupo de agentes que adota tal especificação.

Uma EO é composta por um conjunto de Especificações Estruturais (EE), um conjunto de Especificações Funcionais (EF) e um conjunto de Especificações Deônticas (ED).

Os níveis individual, social e coletivo de uma Especificação Estrutural (EE) podem ser definidos a

partir de três conceitos, que incluem papeis, relações entre papeis e grupos.

Um papel é um conjunto de restrições comportamentais que um agente aceita ao entrar em um grupo. Por sua vez, ao considerarmos o nível social, verificamos que os papeis estão relacionados a outros papeis representando relações impostas às interações entre os papeis.

Quanto ao nível coletivo, um grupo representa um conjunto de agentes com afinidades maiores e objetivos mais próximos. Também, um grupo diferencia-se de uma sociedade justamente por agir como se fosse um único agente cujo comportamento é direcionado a objetivos comuns.

Uma Especificação Funcional (EF) é constituída por um conjunto de esquemas sociais, além de uma relação de preferência entre as missões. A noção fundamental para os esquemas sociais, e conseqüentemente para a especificação funcional, é a noção de meta global. Uma meta global representa um estado do mundo que é desejado pelo SMA.

A cada meta global é associada uma combinação de três valores que indicam: seu nível de satisfatibilidade (atingida, não atingida ou inatingível), seu nível de alocação (comprometido e não comprometido), e seu nível de ativação (permitido e impedido).

O valor inicial de uma meta pode ser: não atingida, não comprometido e impedido, sendo que o seu valor vai sendo alterado no decorrer do funcionamento do sistema.

Uma EF inclui um conjunto de esquemas sociais e a relação de preferência entre missões desses esquemas. Também, pode ser entendida como um SMA que normalmente alcança suas metas globais, isto é, define como essas metas são decompostas por planos e distribuídas aos agentes por missões.

Um Esquema Social (ES) incorpora inicialmente o conjunto de metas, o conjunto de planos que constrói a árvore de decomposição de metas, o conjunto de missões, uma função que determina o conjunto de metas de cada missão, e finalmente, uma função que determina o número (mínimo e máximo) de agentes que devem se comprometer com cada missão.

Um ES é constituído, no nível individual, por missões, que podem ser entendidas como um conjunto coerente de metas globais que pode ser atribuído a um agente através de um de seus papeis. Nesse caso, o agente que se compromete com uma missão é responsável pela satisfação de todas as metas dessa missão.

No nível coletivo, a decomposição de metas é realizada através de planos, responsáveis por indicar uma forma de satisfazer uma meta. Um plano é definido como uma tupla de quatro paridades, incluindo-se a super-meta do plano, o operador do plano, a taxa de sucesso do plano, e a seqüência de sub-metas.

Para relacionar a EE com a EF torna-se necessário estabelecer uma terceira especificação, definida como Especificação Deôntica (ED), e que será responsável por estabelecer quais as missões que um papel tem permissão ou obrigação poderá se comprometer.

A ED de uma organização inclui um conjunto de permissões e um conjunto de obrigações. As permissões determinam que um agente que possui um determinado papel deve se comprometer com uma determinada missão.

Uma obrigação estabelece que um agente que possui um determinado papel é obrigado a se comprometer com a missão em um determinado período de tempo.

Uma EnO é formada por uma finalidade que deve ser mantida por um conjunto de agentes que instanciam a EO.

Por sua vez, o estado de uma EnO, em um determinado instante do tempo, deve incluir a finalidade da entidade, o conjunto de identificadores dos agentes que pertencem a essa entidade, o conjunto dos grupos criados, as especificações dos grupos, os sub-grupos dos grupos.

Também deve incluir, o conjunto de papeis que cada agente está assumindo no instante de tempo considerado, o conjunto de ES criado, a especificação dos ES criados e também quais os grupos que criaram o ES, o conjunto de missões com que cada agente está comprometido, e os estados das metas dos esquemas.

2.3 Modelo $\mathcal{M}$OISE$^{+}_{EDU}$ para Organização de EDI

O aspecto generalista proposto pelo modelo $\mathcal{M}$OISE^{+} apresentado por Hübner [4], que permite que este modelo possa ser aplicado a modelagem de qualquer tipo de organização, no contexto da aplicação neste trabalho, nos leva a estabelecer uma versão adaptada ao caso educacional, que será definida a partir deste ponto como $\mathcal{M}$OISE$^{+}_{EDU}$. Para uma melhor compreensão deste modelo, a tab.1 apresenta as correspondências entre o modelo de curso e o SMA.

Tabela 1. Correspondência entre o modelo de curso e o SMA.

Modelo de Curso	SMA
estrutura curricular do curso	especificação organizacional
relação de docentes com seus comprometimentos	entidades da organização
docentes	agentes
cargos ocupados pelos docentes	papeis
áreas, módulos e unidades curriculares, turmas	grupos
programas das unidades curriculares	planos
itens dos programas das unidades curriculares	missões
relação de cargos dos docentes, áreas, unidades curriculares	especificações estruturais
relação de programas e itens dos programas das unidades curriculares	especificações funcionais
relação descomprometimentos de docentes com as áreas e unidades curriculares	especificações deônticas

Exemplos de modelagem da EE, EF, ED e EnO de EDI podem ser consultados em Duarte et al. [3], sendo que este trabalho mostrará a seguir como a reorganização de EDI está sendo especificada utilizando-se o modelo proposto, em função das necessidades impostas por novas reformulações curriculares e do corpo docente.

2.4 Reorganização de EDI

Esta seção se propõe, com base na proposta de reorganização de Hübner [4] para um SMA, a estabelecer um modelo adequado para que este tenha condições de readaptar-se a novos problemas e situações. Passam, então, a adquirir uma capacidade extra de adaptação à alterações na sua organização, aceitando modificações produzidas em seu contexto de atuação, ou então, para permitir a otimização de seu funcionamento.

A proposta de uma metodologia de reorganização que será apresentada a seguir, baseia-se na alteração dos processos definidos para a definição de um modelo de organização fundamentado em SMA. Isto quer dizer que se pretende produzir alterações, motivadas por necessidades específicas da organização (em nosso caso, educacional), no estado da EO e de seu instanciamento EnO.

Ao considerar-se a necessidade de produção da reorganização das especificações já apresentadas, e entre as possibilidades de mudanças de estado dessas, encontram-se:

- Adoção de um novo papel.
- Remoção de um tipo de grupo.
- Alterações das missões com que os agentes estão comprometidos.
- Alterações dos papeis com que os agentes estão comprometidos.
- Entrada de um novo agente na sociedade.
- Decisão de repartir com outros agentes, os papeis de um determinado agente que está comprometido com diversos papeis (surge muitas vezes motivada pela necessidade de economizar recursos da organização).
- Alterações na estrutura.
- Alterações no funcionamento da sociedade.
- Alterações nas relações deônticas.
- Alterações nas descrições das missões.
- Alterações do período de tempo em que um agente estará comprometido com um papel ou conjunto de missões.
- A visão subjetiva que os agentes possuem das tarefas.
- Ramificação de tarefas e suas conseqüentes realocações aos agentes;
- Alterações nas restrições temporais previamente definidas, fazendo com que um determinado papel deixe de existir em um determinado período de tempo, voltando a existir passado algum tempo, ou sendo substituído por um novo papel.

Entre os tipos de reorganização, definem-se:

- A reorganização já está prevista na EO.
- Apesar de não se conhecer previamente quando a reorganização irá acontecer, o processo que altera a organização é conhecido.
- Não há qualquer controle explícito por parte do sistema para o processo de reorganização.

Por outro lado, entre as formas de produção de uma reorganização definem-se:

- Um agente ou vários agentes do sistema realizam uma etapa do processo de reorganização, podendo ocorrer neste caso, inclusive, uma espécie de auto-reorganização.
- Um usuário do SMA realiza uma etapa do processo de reorganização.

Hübner [4] também define as três etapas empregadas em um processo de reorganização, que incluem:

- Identificação de um situação onde a organização corrente não satisfaz mais o SMA (monitoração), construção de um conjunto de alternativas para a organização corrente (projeto).
- Seleção de alternativas apresentadas pela etapa de projeto, incluindo a definição de critérios de análise de propostas
- Alteração da organização (seleção e implantação).

2.5 Modelo $\mathcal{M}$OISE$^{+}_{EDU}$ para Reorganização de EDI

2.5.1 Reorganização de Docentes

A fig.1 apresenta uma reorganização dos docentes comprometidos com as unidades curriculares do Módulo Geral 1 (MG1) de EDI. Note-se que no diagrama ocorrem três casos distintos com relação aos agentes comprometidos com cada unidade curricular: o docente mantém o comprometimento anterior, o docente deixa de estar comprometido ou surge um novo comprometimento com a unidade curricular. Esta reorganização é necessária, no mínimo ao início de cada semestre, devido as alterações no corpo docente, que podem ser provocadas pelos seguintes fatores: saída de docentes substitutos ao final de seus contratos temporários, saída de docentes afastados temporariamente para a realização de cursos ou que que estejam em licença, saída de docentes provocada por aposentadorias, entrada de novos docentes substitutos ou efetivos, necessidades de realocação produzidas por questões pedagógicas ou de interesse pessoal.

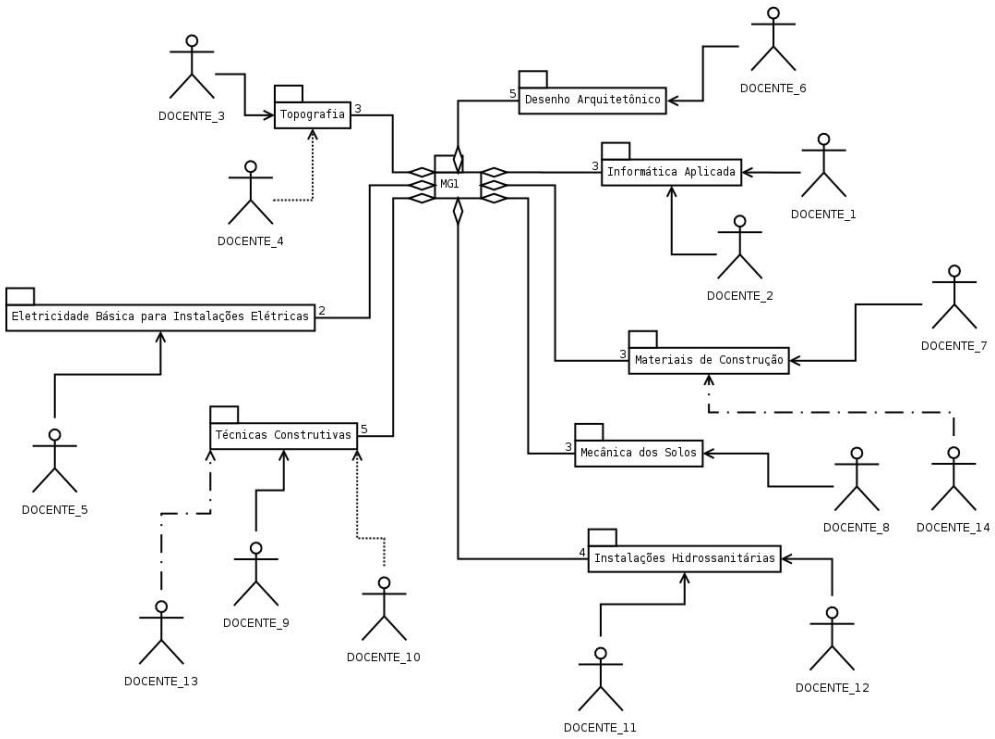


Figura 1 – Comprometimentos, descomprometimentos e novos comprometimentos de docentes com as unidades curriculares do MG1.

A simbologia adotada utiliza setas para a definição dos comprometimentos dos docentes, sendo que se a linha for contínua, sinaliza-se que o comprometimento está mantido. Caso o estilo de linha seja pontilhado, está caracterizada a situação de um descomprometimento do docente com a unidade

curricular. Neste caso, o docente não será responsável pela unidade curricular em um determinado semestre, por exemplo. A situação que indica um novo comprometimento será assinalada com uma linha em estilo centrado (traço-ponto-traço). Por exemplo, com relação à unidade curricular de Técnicas Construtivas, o DOCENTE_9 mantém seu comprometimento, enquanto que o DOCENTE_10 deixa de estar comprometido. Por sua vez, o DOCENTE_13 passa a ter um comprometimento com a unidade curricular.

Os grupos de unidades curriculares são interligados ao grupo que define o módulo (MG1, para o primeiro semestre do curso), por uma relação de herança, assinalada por um losango na extremidade que indica o grupo principal, e um número na ligação com o ícone de grupo da unidade curricular. Este número será utilizado para indicar a multiplicidade de carga horária para cada unidade curricular. Por exemplo, a unidade curricular de Técnicas Construtivas terá cinco horas-aula semanais. Esta notação de herança é adotada também pela *UML* [6] e também é definida para o modelo $\mathcal{M}$OISE$^+$ por Hübner [4].

2.5.2 Reorganização de Turmas

A reorganização de turmas utilizando o modelo $\mathcal{M}$OISE$^+_{EDU}$ pode ser realizada conforme exemplificado na fig.2, para o caso da unidade curricular de Materiais de Construção, ofertada no MG1 de EDI.

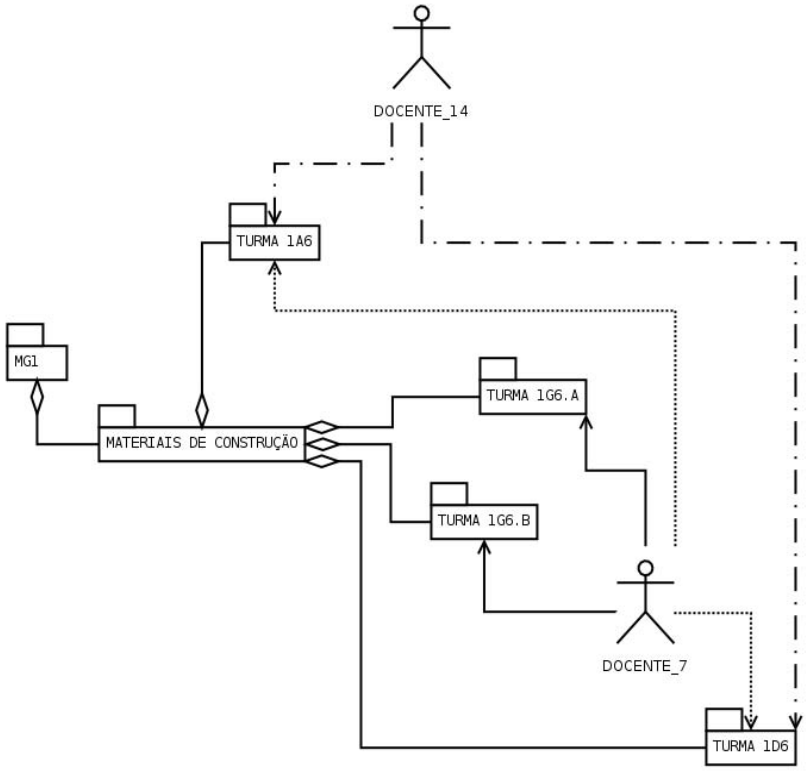


Figura 2 – Reorganização de comprometimentos de docentes com as turmas da unidade curricular de Técnicas Construtivas no MG1.

O estudo de caso foi realizado para a situação analisada para o semestre 2007/2, sendo que o agente DOCENTE_7, anteriormente era o único docente comprometido com a unidade curricular (no semestre 2007/1). Após a reorganização, o agente DOCENTE_14 foi incluído para um novo comprometimento (estilo de linha centrado), sendo que será responsável pelos grupos TURMA 1A6 (turno da manhã) e TURMA 1D6 (turno da tarde).

Por sua vez, o agente DOCENTE_7 estará comprometido somente com os grupos TURMA 1G6.A e TURMA 1G6.B (turno da noite), abandonando os comprometimentos com os grupos TURMA 1A6 e TURMA 1D6 (estilo de linha pontilhado), que passam a ser responsabilidade do agente DOCENTE_14.

A visualização da nova situação é útil ao agente COORDENADOR_PEDAGÓGICO do curso, pois permite identificar visualmente os comprometimentos dos agentes DOCENTES envolvidos com as unidades curriculares, com suas respectivas turmas. É importante salientar, que o histórico das alterações realizadas não se perde, ajudando na compreensão da nova situação criada no curso, ao ingressar em um novo semestre letivo. Neste caso, é importante que o curso tenha em sua biblioteca virtual todos os arquivos gerados a cada semestre letivo, facilitando inclusive a inspeção pelos organismos competentes, o que passa a ser bastante útil no caso de uma avaliação do curso pelo MEC.

2.5.3 Reorganização de Conteúdos nas Unidades Curriculares

Com relação à reorganização da especificação funcional, que relaciona os planos de uma unidade curricular, incluindo suas metas e missões, a fig.3 ilustra um exemplo de diagrama para o modelo $\mathcal{M}\text{OISE}^{+}_{\text{EDU}}$.

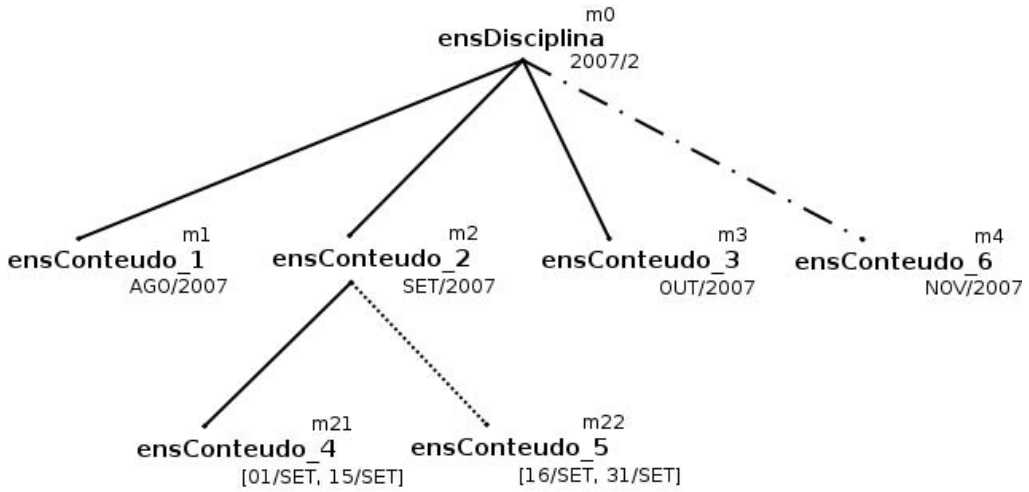


Figura 3 – Reorganização da especificação funcional de uma unidade curricular.

O diagrama mostra que as metas *ensConteudo_1* (a ser atingida em AGO/2007), *ensConteudo_2* (a ser atingida em SET/2007), *ensConteudo_3* (a ser atingida em OUT/2007), *ensConteudo_4* (a ser atingida no período de 01/SET até 15/SET), estão mantidas no semestre 2007/2 (usou-se como convenção, o estilo de linha contínua). Por sua vez, a meta *ensConteudo_5* (a ser atingida no período de 16/SET até 31/SET) deixa de fazer parte do conjunto de de metas a serem atingidas (usou-se como convenção, o estilo de linha pontilhado).

Observe-se que esta alteração pode ser motivada por diversos fatores, que podem incluir, por exemplo, uma reformulação curricular prevista para o semestre reorganizado. Uma nova meta (*ensConteudo_6*, a ser atingida em NOV/2007) é incluída (usou-se como convenção, o estilo de linha centrado), devendo ser atingida pelo(s) agente(s) que estiverem comprometidos com a meta *ensDisciplina*, em 2007/2.

Com relação aos comprometimentos de docentes com as missões, este pode ser definido a partir de diagrama de especificação deôntica, ilustrado na fig.4. Neste caso, o agente DOCENTE_A, mantém seus comprometimentos com as missões *m1* e *m21* (setas em estilo de linha contínua), deixando de estar comprometido com as missões *m22* e *m3* (setas em estilo de linha pontilhado), enquanto que o

agente DOCENTE_B, passa a ter comprometimentos com as missões *m3* e *m4* (setas em estilo de linha centrado).

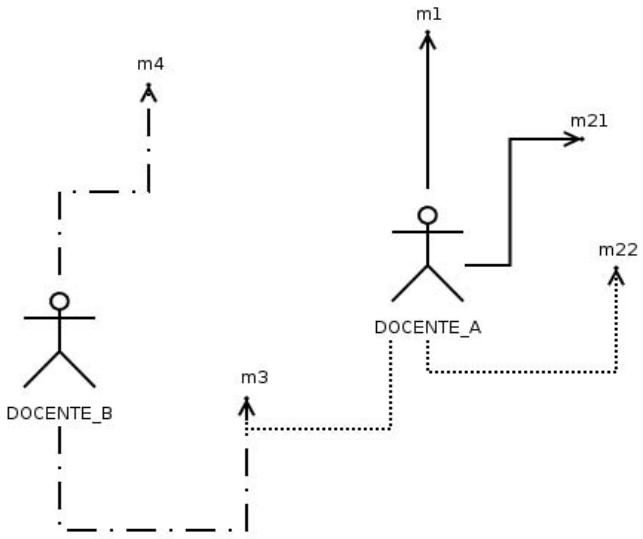


Figura 4 – Reorganização da especificação deôntica de uma unidade curricular.

3 CONCLUSÃO

Neste trabalho apresentou-se um procedimento metodológico, fundamentado na modelagem da organização e reorganização de cursos, sendo mediada pelo uso da tecnologia de sistemas multiagentes. O objetivo principal na adoção da metodologia proposta consiste na obtenção de uma melhoria na qualidade de ensino, a partir de um melhor entendimento do funcionamento do curso modelado.

Pretende-se confirmar que este objetivo pode ser atingido, a partir do momento em que as tecnologias e ferramentas computacionais propostas, estão sendo utilizadas pelos docentes e coordenadores de curso, na reestruturação de suas idéias e atividades didático-pedagógicas e de coordenação.

É importante considerar a idéia de que o uso de tais tecnologias e ferramentas computacionais, especificadas neste trabalho, não consiste em uma obrigatoriedade de seu uso para que os objetivos acima citados possam ser atingidos. Por sua vez, afirmamos que o enfoque principal deste trabalho reside na proposta de adoção da metodologia para a organização e reorganização de cursos, com base em sistemas multiagentes, com utilização do modelo $\mathcal{M}$OISE^{+}, sugerido por Hübner [4][5], adaptado ao caso educacional neste trabalho, passando a ser denominado $\mathcal{M}$OISE$^{+}_{\text{EDU}}$.

Na atual fase do desenvolvimento deste trabalho, pretende-se envolver o máximo possível os docentes e coordenadores de EDI, no desenvolvimento orientado de protótipos específicos (*standalone* ou na Internet) para a aplicação da metodologia de modelagem proposta. Os protótipos em desenvolvimento já estão sendo utilizados nas reestruturações do curso, com o objetivo de encorajar os docentes a participarem mais ativamente na definição de novas metodologias de ensino, para a obtenção de melhores resultados nas atividades de ensino-aprendizagem.

Note-se, ainda, que o ideal seria que cada docente responsável pelas áreas e unidades curriculares, realizasse a construção dos diagramas apresentados, com orientação de um engenheiro de conhecimento. Deveriam considerar, adicionalmente, a possibilidade de inclusão de itens

multimídia que seriam complementares aos diagramas, tais como imagens e vídeos relacionados aos itens constituintes das bases científicas e tecnológicas, relativas à área de construção civil. Isso possibilitaria a conversão de dados educacionais complexos em formulários visuais que comportam recursos hipermídias que complementariam as informações apresentadas pelos diagramas do modelo $\mathcal{M}$OISE$^{+}_{EDU}$. Pretende-se trabalhar com esta possibilidade em futuros trabalhos a serem desenvolvidos.

A modelagem de outros cursos, além de EDI, também está sendo experimentada no CEFET-RS, incluindo a implantação de um novo curso de EDI, na modalidade ensino integrado com as disciplinas do ensino médio.

Um novo curso de graduação em Engenharia Civil, a ser ofertado pelo CEFET-RS, provavelmente, a partir de 2008/1, também está sendo modelado utilizando-se o modelo $\mathcal{M}$OISE$^{+}_{EDU}$, de forma a obtenção de um curso adequado às exigências do mercado de trabalho e adaptado às exigências legais, desde o seu nascimento. Os diagramas apresentados neste trabalho estão possibilitando a identificação da viabilidade de implantação do curso proposto, em função de uma análise do corpo docente existente no CEFET-RS.

A seqüência deste trabalho também deverá incluir a implementação de um editor gráfico especificamente desenvolvido para a construção dos diagramas propostos para o modelo $\mathcal{M}$OISE$^{+}_{EDU}$. Pretende-se que através deste editor, de utilização semelhante ao *Cmap Tools* [7], o usuário (docente ou coordenador de curso) construa os diagramas, e o aplicativo gere automaticamente a codificação *XML* [2], de forma a permitir o armazenamento das informações educacionais modeladas em uma base de dados que poderá ser visualizada através da Internet. Espera-se, que a partir de consultas a esta base de dados, seja possível aos docentes e coordenadores uma análise mais adequada dos desenvolvimentos pedagógicos e curriculares envolvidos.

REFERÊNCIAS

[1] Briot, J. P.; Demazeau, Y. *Principes et Architecture des Systèmes Multi-Agents*. Paris: Hermes, 2002.

[2] Dick, K. *XML: A Managers's Guide*. Boston: Addison-Wesley Professional, 2003.

[3] Duarte, G. D.; Costa, A. C. da R.; Carneiro, M. L. F. Diagramas $\mathcal{M}$OISE^{+} para Organização Curricular do Curso Técnico em Edificações no CEFET-RS. In: CICLO DE PALESTRAS SOBRE NOVAS TECNOLOGIAS PARA A EDUCAÇÃO, 9., Porto Alegre, 2007.

[4] Hübner, J. F. *Um Modelo de Reorganização de Sistemas Multiagentes*. São Paulo: Escola Politécnica da Universidade de São Paulo, tese, 2003.

[5] Hübner, J. F.; Sichman, J. S.; Boissier, O. $\mathcal{M}$OISE^{+}: Towards a Structural, Functional, and Deontic Model for MAS Organization. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, Bologna, p.501-502, 2002.

[6] Miles, R.; Hamilton, K. *Learning UML 2.0*. Cambridge: O'Reilly Media, 2006.

[7] Novak, J. D. Using concept maps to facilitate classroom and distance learning. SCUOLA & CITTÀ. Roma: La Nuova Italia, v.2, p.112-114, 2002.

[8] Sastre, J. R. de S. et al. *Curso Técnico de Edificações – Construção Civil – Projeto de Reforma*. Pelotas: CEFET-RS, 2004.

[9] Vázquez-Salceda, J.; Dignum, V.; Dignum, F. Organizing Multiagent Systems. AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS. Hingham-MA: Kluwer Academic Publishers, v.11, i.3, p.307-360, november, 2005.

Infraestructura para laboratorios de acceso remoto

Lopez Luro, Francisco - Bertogna, Leandro - Sanchez, Laura - Rodriguez, Jorge - Del Castillo, Rodolfo

Universidad Nacional del Comahue, Departamento de Ciencias de la Computación Neuquén, Argentina
{flopez,mlbertog,lsanchez,jrodrig,rolo}@uncoma.edu.ar

Resumen

Las nuevas tecnologías de Internet permiten el uso de sistemas de software distribuido para el acceso en forma remota a laboratorios físicos y virtuales, para llevar a cabo actividades de investigación experimental a distancia. A partir de los marcos teóricos que definen los estructurantes de los laboratorios, los implementadores de laboratorios de acceso remoto deben diseñar y desarrollar arquitecturas de servicios que permitan un acceso flexible y controlado. Este trabajo define una infraestructura para la implementación de aplicaciones para el acceso remoto a laboratorios físicos y virtuales y para la gestión de los mismos. Se presentan conceptos que permiten extender la modalidad tradicional de conexión con un dispositivo, para llevar adelante actividades prácticas de laboratorio colaborativas y concurrentes.

1. Introducción

Las nuevas tecnologías de Internet y el aumento de la velocidad de los medios de comunicación digital permiten el uso de sistemas de software distribuido para el acceso en forma remota a laboratorios virtuales o físicos, para llevar a cabo actividades de investigación experimental a distancia.

Un laboratorio de acceso remoto (LAR) es "un espacio de trabajo electrónico para la colaboración y experimentación en investigación u otras actividades creativas, para la generación y distribución de los resultados de investigación utilizando tecnologías de información distribuidas"[10].

Mediante los laboratorios de acceso remoto, recursos virtuales y físicos, escasos y dispersos, pueden ponerse a disposición de mayor cantidad de investigadores y estudiantes.

A partir de los marcos teóricos que definen los estructurantes de los laboratorios, los implementadores de LAR deben diseñar y desarrollar arquitecturas de servicios que permitan el acceso a sus recursos de manera flexible y controlada. Las tecnologías de comunicación para el acceso remoto a los laboratorios deben ser variadas e intercambiables, pero siempre respetando la semántica de la información que se transmite desde y hacia los recursos. De esta manera se logra un nivel de independencia entre la información que generan o consumen los recursos y el medio utilizado para ello, que permite agregar nuevos recursos a los LAR sin modificar sus arquitecturas de servicios actuales, sino extendiendo y adaptándose a las mejores tecnologías de comunicación.

Por su parte, las aplicaciones que acceden a los recursos de los LAR deben poder adaptarse a nuevos servicios y a los cambios tecnológicos de servicios existentes. Esto se logra construyendo aplicaciones

modulares y extensibles, sin limitar el diseño a las tecnologías de comunicación o a los métodos de acceso utilizados para conectarse a los recursos.

Un Framework Orientado a Objetos (FOO) es un diseño reusable de una aplicación o subsistema, representado por un conjunto de clases abstractas, y la forma en que estas clases colaboran [12, 13]. Este diseño representa una aplicación semi-completa para una clase especifica de software, que puede ser especializada para producir aplicaciones a medida.

Este trabajo describe una infraestructura para la implementación de aplicaciones para el acceso remoto a laboratorios y para la gestión de los mismos. Para la construcción de aplicaciones de usuario, se diseñó un FOO basado en Java[3] que permite la construcción de aplicaciones altamente adaptables a los cambios tecnológicos y desacopladas de las tecnologías de comunicación utilizadas para el acceso, obteniendo herramientas reusables para el acceso a distintos laboratorios, sin importar las tecnologías por las cuales se acceda, además de facilitar la adición y el acceso a nuevos servicios.

Para los servidores de laboratorios, se diseñó una arquitectura de servicios que provee a múltiples usuarios distantes entre sí, el acceso concurrente, colaborativo y gestionado dinámicamente, a recursos físicos y virtuales provistos por una o más organizaciones.

En la sección siguiente se presenta la arquitectura general. En la sección 3 se presentan los conceptos estructurantes y la definición de un recurso. En la sección 4 se presenta el Framework para la creación de aplicaciones Cliente para acceder a recursos remotos. La sección 5 describe las mejoras implementadas en la arquitectura del Servidor de Recursos. La sección 6 presenta un prototipo funcional de la infraestructura.

2. Arquitectura general

Los aportes que se encuentran en este trabajo están insertos en la arquitectura de la figura 1.

Los componentes de dicha arquitectura comprenden:

El "Servidor de labs", que funciona como un portal web para los clientes (alumnos, docentes y administradores), permitiendo que estos inicien una sesión segura, accedan a contenidos teóricos varios, planifiquen actividades prácticas e inicien su participación en un LAR. El "Servidor de Recursos (SR)", que provee acceso a los recursos físicos y virtuales en una o más organizaciones, es decir, está conectado físicamente a los recursos. El "Servidor de Aplicación de Políticas" es el encargado de hacer efectivas las diferentes políticas sobre los SR para permitir que cada cliente luego de ser autenticado pueda acceder a los distintos recursos del laboratorio con los permisos que corresponda. El Servidor LDAP actúa como base de datos de autenticación para alumnos, docentes y administradores para acceder al portal y por lo tanto a los laboratorios.

Se puede encontrar una visión más detallada en [1, 2], donde se presentan aspectos y mejoras del diseño de aplicaciones Cliente que acceden al laboratorio, y agregados de funcionalidad a los servicios del LAR para dar soporte a actividades de e-learning colaborativas .

En particular, este trabajo se enfoca en los Clientes de laboratorios y en el concepto y la operación de lo que llamamos Distribuidor y Manejador de Conexiones (DMC) y Concentrador de Conexiones (CDC). Estos últimos, proveen a los Servidores de laboratorios la funcionalidad necesaria para poder gestionar distintos laboratorios al mismo tiempo, permitir el acceso a grupos de alumnos por laboratorio y conectarse para trabajar de manera concurrente y colaborativa, permitiendo opcionalmente, la presencia de un facilitador que altere la dinámica de trabajo durante la práctica en forma remota.

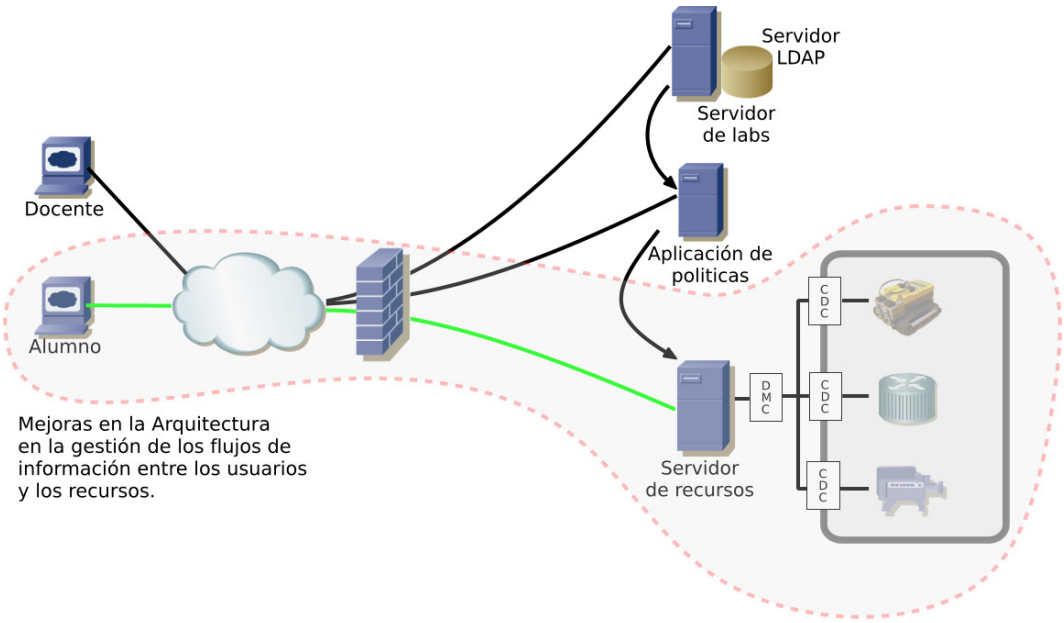


Figura 1: Arquitectura general

3. Recursos remotos físicos y virtuales

Utilizaremos el término recurso en su sentido más amplio, como una entidad real o virtual que puede producir información de cualquier naturaleza en función de un requerimiento de entrada y que tiene un estado asociado. Estos recursos pueden ser estáticos o dinámicos y su acceso puede realizarse en forma sincrónica o asincrónica.

Ejemplos de estos recursos pueden ser páginas web, sensores físicos de cualquier naturaleza, dispositivos que generan flujos multimedia, un generador de números aleatorios de alta precisión, una máquina virtual de pc, etc.

Existen muchos recursos hoy en la Internet que son gestionados y accedidos de manera eficiente. Muchas de las tecnologías y herramientas de groupware para el acceso a dichos recursos han demostrado su efectividad y no necesitan ser reemplazadas en el corto plazo. El protocolo de aplicación HTTP por ejemplo, implementado por los navegadores web y los servidores web, es utilizado masivamente para permitir el acceso a contenido Hipertextual (HTML) de forma simple y efectiva.

Este trabajo complementa dichas tecnologías y aplicaciones instauradas en la comunidad de usuarios de Internet, permitiendo el acceso a recursos de variada naturaleza, a través de interfaces bien definidas, para ser utilizados en actividades de investigación y educación a distancia. Ejemplos de estos recursos pueden ser “dispositivos de hardware especial", “interpretes de comandos de texto", “microscopios electrónicos", “robots", etc. Estos recursos a veces escasos o no disponibles en todos los laboratorios generan flujos de información, los consumen como datos de entrada, o ambas cosas.

Se define entonces para este trabajo el concepto de recurso como una entidad que puede ser Productora (Productor) de información, Consumidora (Consumidor) de información, o ambas. Además, un recurso dispone de cero o más operaciones que modifican su estado y funcionamiento, que deben poder ser ejecutadas sobre este a distancia y en un ambiente controlado y seguro.

Las posibilidades de tener acceso a un recurso en particular dependen de la existencia de una interfaz

de entrada/salida bien documentada y del ancho de banda necesario para transmitir los datos desde el recurso hacia el usuario y viceversa, que puede variar desde un simple flujo de caracteres hasta audio y video de alta calidad.

El agrupamiento lógico de los recursos dentro de un “laboratorio de acceso remoto" facilita la administración y disposición para las distintas actividades que quiera llevar adelante una institución.

4. Framework para la construcción de clientes de LAR

Las aplicaciones que permiten el acceso a recursos dentro de LARs, sigue claramente un modelo Cliente-Servidor [17], en el cual el proveedor de los servicios, quien crea y dispone los recursos reales o virtuales en sus laboratorios, implementa dichos servicios mediante alguna tecnología de sistemas distribuidos que dé soporte al acceso remoto a sus recursos.

Ejemplos de estas tecnologías de comunicación pueden ser RMI, CORBA, Remote Procedure Calls (RPC), Web Services, el protocolo HTTP, o un socket TCP entre dos pares. Cada una de ellas, tiene sus ventajas y desventajas en el momento de implementar un servicio, publicarlo y permitir el acceso.

Por ejemplo, para el acceso a un intérprete de texto los Web Services no proveen el grado de interacción suficiente para generar la teleprescencia deseada en un recurso con tal interfaz

Ante la gran diversidad de recursos y tecnologías de comunicación existentes, se plantea la necesidad de construir aplicaciones Cliente que tengan previsto desde su diseño, la posibilidad de intercambiar fácilmente la tecnología de comunicación utilizada para acceder a los recursos.

4.1. Diseño del Framework

En el contexto del diseño de software, un framework puede ser visto como la materialización concreta de familias de patrones de diseño[16] para resolver problemas de un dominio de aplicación en particular, mientras que por otro lado, los patrones pueden ser vistos como elementos abstractos microarquitecturales de los frameworks que documentan y motivan la semántica del framework de una manera efectiva.

El lenguaje utilizado para la implementación del Framework es Java [5]. Esta decisión está basada fundamentalmente en que Java es un lenguaje orientado a objetos, multiplataforma y permite la carga de clases de manera dinámica, facilitando el desacoplamiento de la lógica de la aplicación, la interfaz de usuario y la tecnología de comunicación para acceder a un recurso específico. El uso de la tecnología de Applets Java simplifica las tareas de distribución y actualización del software[6].

La figura 2 muestra el diagrama de clases en UML del framework diseñado.

4.2. Descripción general del diagrama de clases del Framework

El usuario del framework, representado en el diagrama por la clase *Cliente*, es la aplicación final que provee la lógica de aplicación y la interfaz de usuario (UI). Esta clase *Cliente* usará los servicios que proveen las clases *Laboratorio*, *Recurso* y *Operacion*, y las interfaces *Productor*, *Consumidor* para poder acceder a los recursos remotos de manera transparente.

La clase *Laboratorio* es una colección de *Recursos*, y es el punto de entrada al framework desde el punto de vista de una aplicación final. Desde la clase *Laboratorio*, la clase *Cliente* puede acceder a los distintos recursos que componen un LAR. Cada clase *Recurso* consiste de una colección de *Atributos*

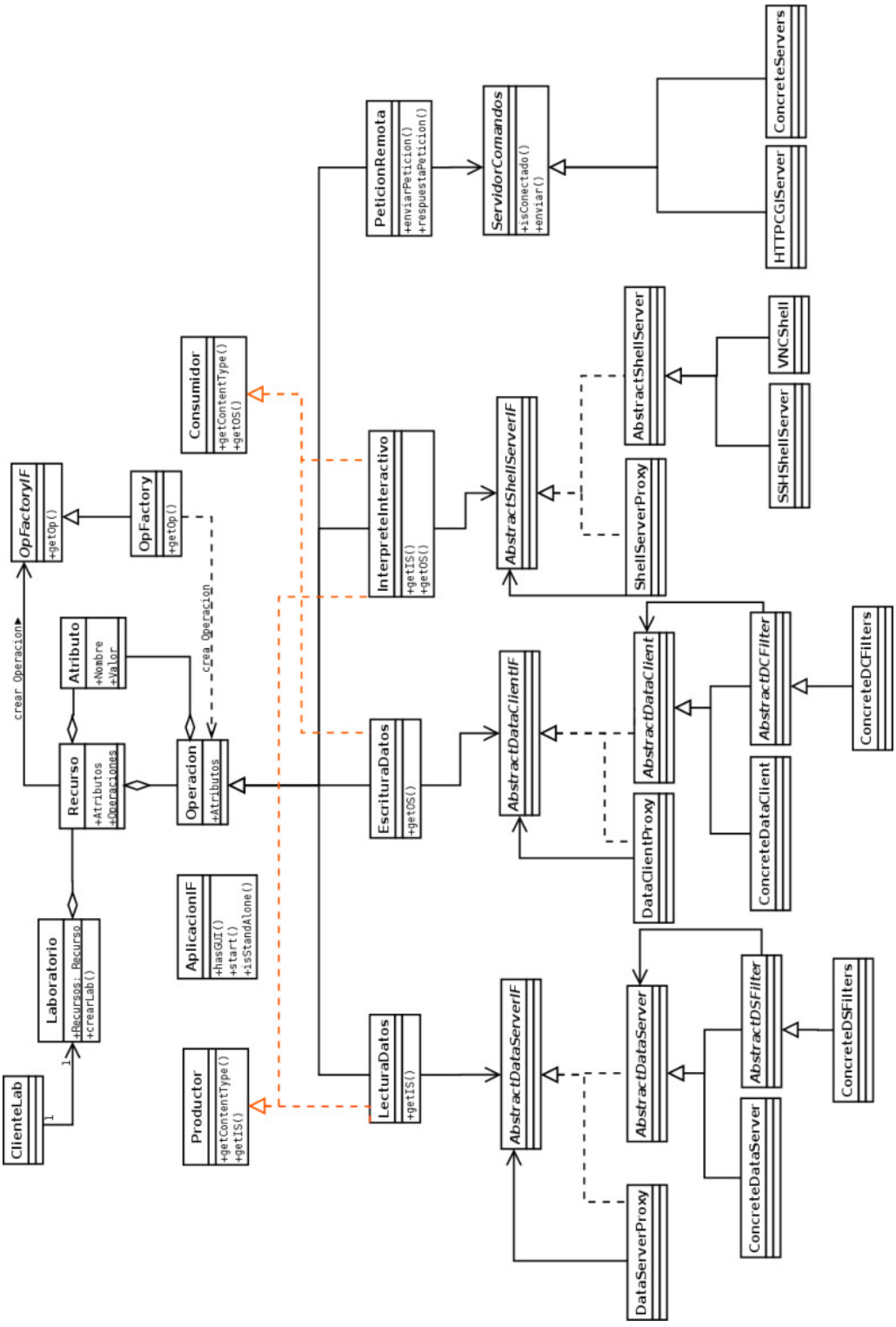


Figura 2: Diagrama de clases

y de *Operaciones*, donde los atributos son simplemente pares (nombre,valor) que permiten enumerar cualquier tipo y cantidad de atributos para un recurso, mientras que las operaciones pueden ser una de las cuatro subclases de la figura 2, *LecturaDatos*, *EscrituraDatos*, *InterpreteInteractivo* o *PeticionRemota*.

Existen tres roles, representados como interfaces Java en el diagrama de clases, que caracterizan el comportamiento de las distintas operaciones. La interfaz *Productor* será implementada por aquellas operaciones que necesiten leer datos desde un *Recurso*, análogamente la interfaz *Consumidor* será implementada por aquellas operaciones que envíen datos a un *Recurso*.

Asociadas a cada subclase de *Operacion*, están las interfaces de implementación. Dichas interfaces desacoplan totalmente las implementaciones concretas con las cuales se accede al recurso remoto. De esta manera se pueden reemplazar las implementaciones respetando las interfaces sin la necesidad de modificar la aplicación final e incluso en tiempo de ejecución de la misma.

Para poder acceder a un LAR concreto se necesita conocer la información sobre los recursos que este ofrece y las tecnologías necesarias para acceder a dichos recursos. Por ejemplo, un laboratorio puede ofrecer un recurso "sensor" que representa lecturas de un proceso en un laboratorio real y permite el acceso a dichas lecturas mediante un simple socket TCP. El Cliente del framework accederá a la información que genere dicho sensor mediante una operación de lectura de datos sin conocer en concreto que la lectura esta siendo realizada con una determinada tecnología (clase *LecturaDatos*).

Según sea necesario, se irán agregando implementaciones concretas, que luego podrán ser utilizadas por cualquier diseñador de una aplicación Cliente que tenga acceso a las clases compiladas o el código fuente. Técnicamente, estas implementaciones se denominan hot-spots o puntos de extensión en el Framework.

Se debe notar que las implementaciones que permiten comunicarse mediante algún tipo de tecnología no son parte del framework, son Clases independientes que implementan las interfaces correctas del framework para poder ser usadas dentro del mismo. Estas pueden ser cargadas en forma dinámica durante la ejecución del framework, lo que agrega flexibilidad y capacidad de adaptación de la aplicación en tiempo de ejecución.

Como se mencionó, es necesario conocer con qué tecnología se implementó el servicio para acceder a la información que genera un recurso determinado. Cada tecnología necesita distintos parámetros para su correcta configuración, el enfoque que se adoptó es el de disponer de una descripción genérica y flexible, en formato XML, del recurso y sus operaciones, permitiendo abarcar las descripciones de tecnologías actuales así como de tecnologías nuevas o aún desconocidas.

La descripción genérica de un recurso en XML se corresponde con las clases *Recurso*, *Operacion* y *Atributo* del diagrama y tiene la siguiente forma:

```
<recurso>
<atributos>
  <atributo><nombre>...</nombre><valor>...</valor></atributo>
 <atributos>
 <operaciones>
  <operacion>
   <atributos>
    <atributo><nombre>...</nombre><valor>...</valor></atributo>
   </atributos>
```

```
    </operacion>
   </operaciones>
  </recurso>
```

El Cliente del framework debe proveer una configuración en XML que describa el laboratorio, para que las clases del framework se puedan instanciar correctamente. Toda la información necesaria para describir completamente cada recurso y construir sus operaciones debe estar en este XML. Por ejemplo, si la comunicación con un recurso determinado se realizará usando un socket TCP, se debe conocer la dirección IP y el puerto en el que se puede establecer la conexión.

Básicamente, solo la implementación concreta posee la información de qué argumentos necesita para ser inicializada correctamente, entonces el Framework durante su construcción, consulta a dichas clases por los atributos necesarios, los extrae del XML y se los pasa a través de un método de inicialización común. Con esta técnica se elimina la necesidad del uso de introspección para la creación de clases desconocidas al momento de compilación. Si faltara alguno de los atributos en el XML se produce un error fatal en el Framework dado que la implementación no podrá ser inicializada correctamente.

La interfaz *AplicacionIF* tiene un propósito especial en el diseño, y es el de poder reusar una aplicación final hecha en Java, como puede ser una terminal SSH o un Cliente VNC de escritorio remoto. Mediante el uso de esta interfaz, el programador puede consultar a una determinada operación de un recurso si esta provee una aplicación final (con GUI propia) independiente y lanzarla en caso afirmativo con los argumentos que necesite. Este enfoque permite adaptar fácilmente aplicaciones existentes a los LARs y reusarlas de manera transparente dentro del framework.

4.3. Patrones de diseño

A continuación se describen los patrones presentes en el diagrama de clases:

- Interface[14]: El patrón interface permite mantener una clase que hace uso de los datos y los servicios de instancias de otras clases independiente de dichas clases mediante el acceso a estas a través de una interfaz[15]. El lenguaje Java dispone de un tipo de dato específico para este patrón, denominado justamente *Interface* y hace exactamente lo que define el patrón de mismo nombre. Mediante la aplicación de este patrón, se logra desacoplar el diseño genérico del framework de las implementaciones concretas que acceden a los recursos reales. Las clases *AbstractDataServerIF*, *AbstractDataClientIF* y *AbstractShellServer* entre otras, realizan este patrón.

- Filter[14]: El patrón filter permite que objetos que realizan diferentes transformaciones y cálculos sobre flujos de datos sean conectados en forma dinámica para componer transformaciones mas complejas. El único requerimiento para estos objetos transformadores es que implementen una interfaz común. El concepto es similar a las tuberías de Unix que procesan flujos de bytes o caracteres en forma separada como filtros. Las clases concretas de filtros pueden ser compuestas en forma dinámica sin necesidad de modificar la aplicación Cliente, con lo cual la aplicación puede adaptarse a situaciones nuevas como por ejemplo que los datos de la red vengan en un formato comprimido. Las clases centrales de este patrón en el diagrama son *AbstractDSFilter* y *AbstractDCFilter*.

- Virtual Proxy[14]: Si la instanciación de un objeto es costosa, y quizás el objeto no sea utilizado, puede ser provechoso retrasar dicha instanciación hasta el momento preciso en que se requiera

de los servicios del mismo. Al disponer de una interfaz igual a la del objeto en cuestión, que simplemente tenga una referencia al objeto y delegue las operaciones requeridas por los clientes en el instante que sucedan, se puede agregar un nivel de indirección en los requerimientos. De esta manera, los clientes no pueden saber en que momento es construido el objeto real que implementa el servicio. A esta técnica también se la denomina "Lazy Instantiation" (instanciación retrasada).
Si la aplicación basada en el framework se debe descargar de la red, como es el caso del LAR, la descarga será muy rápida puesto que solo se transmite el esqueleto del framework y la aplicación cliente, dejando las implementaciones concretas para descargar cuando se realicen las solicitudes de servicio por el usuario final. Las clases *DataServerProxy*, *DataClientProxy*, *ShellServerProxy* implementan en parte este patrón.

5. Acceso controlado y concurrente a los recursos del laboratorio

5.1. Distribuidor y Manejador de conexiones (DMC)

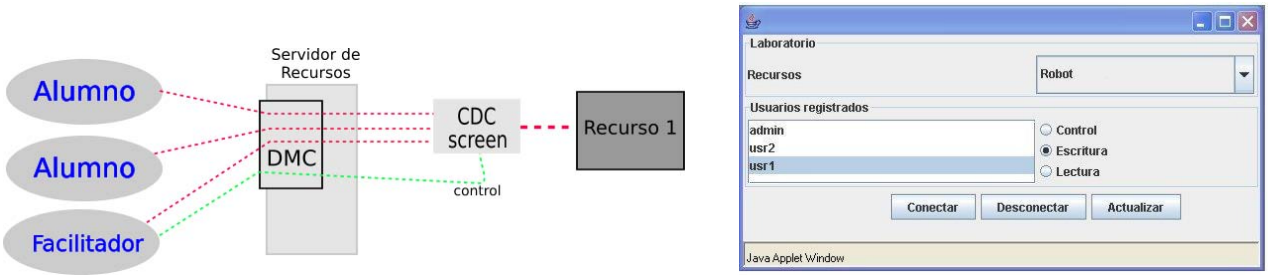
Desde el punto de vista de la institución o empresa, sus recursos deben poder ser accedidos de manera coordinada y controlada, para cumplir con esta restricción, definimos el concepto de "Distribuidor y Manejador de conexiones" (DMC), que es un componente de software que se ejecuta como un servicio de red en el SR y su principal objetivo es controlar el acceso de los clientes a los distintos recursos disponibles. Pueden existir una o más instancias de DMCs en cada SR según sea necesario acceder a distintas clases de recursos.

Como se mencionó anteriormente, los recursos están englobados lógicamente en una organización por laboratorio, por lo tanto, un usuario que quiera acceder a un recurso en particular debe poder hacerlo también a su respectivo laboratorio. Se optó por un modelo de control de acceso a los recursos basado en roles (RBAC) para gestionar el laboratorio, implementado a través de un servidor LDAP. A la hora de acceder al laboratorio, cada usuario tendrá uno o más roles asignados, creados al momento de la construcción del laboratorio específico, que le permitirán realizar determinadas acciones sobre cada uno de los recursos disponibles en el laboratorio.

Cuando un usuario se autentica en el sistema, a través de la interfaz web en el Servidor de Laboratorios, y escoge el laboratorio práctico que va a realizar, el Servidor de Aplicación de Políticas, con la información del usuario y el laboratorio correspondiente, configura los DMC en los Servidores de Recursos afectados para que el usuario pueda acceder. Luego que se han configurado los DMC, queda disponible un archivo XML con las descripciones de los recursos, sus atributos y operaciones, que será usado por la aplicación Cliente del usuario final para configurarse dinámicamente y acceder al laboratorio.

5.2. Concentrador de Conexiones (CDC)

Para poder llevar adelante las prácticas remotas satisfaciendo las características de un entorno colaborativo, se necesita extender el esquema de conexión básico con un recurso, para permitir compartir de manera controlada la información que entra y sale de estos, concentrando todas las conexiones entrantes hacia el recurso y replicando la salida del recurso hacia las distintas conexiones (clientes). Este componente de software lo denominamos Concetrador de Conexiones (CDC), cada clase o tipo de recurso



(a) Esquema de DMC y CDC en un Servidor de Recursos (b) Front-End para gestión de CDC de caracteres

Figura 3:

tendrá su propio CDC y su complejidad variará según la naturaleza del recurso y el control que se quiera ejercer sobre los flujos de datos. Un esquema conceptual de la función de los componentes DMC y CDC se puede ver en la figura 3 se muestran conceptualmente la función del DMC y el CDC relacionados a un recurso y un conjunto de usuarios.

Por ejemplo, en la clase de recursos orientados a caracteres como consolas de texto, donde todo lo que entra y sale de un dispositivo son caracteres, se optó por implementar un CDC utilizando la herramienta de software"GNU Screen"[4]. Este concentrador permite conectar fácilmente las entradas y salidas estándar de varios procesos a uno en particular, que en nuestro caso es un proceso asociado al recurso en cuestión. Además, se puede controlar el flujo de caracteres de cada una de las conexiones entrantes, limitando una conexión por ejemplo a "solo lectura". Entonces el Cliente conectado con "solo lectura" es un simple observador de lo que está entrando al recurso. Esto es útil por ejemplo, para que durante una práctica remota un solo usuario a la vez pueda enviar información hacia el recurso, mientras el resto observa en el mismo instante los cambios.

Debido a que los CDC pueden ser bastante complejos de administrar en tiempo real, se han desarrollado varias herramientas para hacer más amigable su configuración y uso, para aislar al facilitador o docente de la complejidad del concentrador en sí mismo y solamente presentar una interfaz conceptual de las operaciones sobre cada uno de los roles (alumnos) y recursos asociados.

En particular, para el concentrador de conexiones orientadas a caracteres, se ha desarrollado una herramienta denominada "screenController" en lenguaje Python que se comunica con el programa "screen" para modificar en tiempo de ejecución el estado de las conexiones con un recurso determinado. Además se desarrolló un Applet Java (front-end) para interactuar visualmente con el "screenController" (parte derecha de la figura 3) y permitir al docente controlar y modificar remotamente el estado del CDC desde el cliente de LARs.

6. Experiencia de laboratorio de robots

Durante la demostración del prototipo de LAR de programación de robots realizada en TEyET'06 desde la Universidad Nacional del Comahue, en el marco del proyecto "Software para Aprendizaje y Trabajo Colaborativos"[2], un grupo de alumnos participó a distancia y de manera colaborativa en una clase de programación a distancia.

El laboratorio consiste de una etapa teórica y una práctica. El docente realiza una presentación introductoria mediante transparencias a través de la web para explicar el funcionamiento básico del robot y la

forma de trabajo. A continuación, los alumnos concretan la práctica de programación propuesta usando un cliente SSH para conectarse a una terminal remota en el SR.

En el SR, podrán programar el robot y ejecutar dicho programa para ver los resultados concretos en un robot real a través de una camara web via HTTP. Para lograr una experiencia colaborativa que permita a todos los integrantes poder hacer un aporte en el resultado final, y haciendo uso del CDC desarrollado para consolas de textos, los alumnos se conectan a la misma consola de texto, por la cual todos ven una única imagen del estado del recurso, y el instructor define de manera dinámica quien puede escribir en dicha consola en un instante determinado. Durante el ejercicio de programación el/los alumnos reciben la asistencia del docente y de sus compañeros para resolver el problema planteado, permitiendo que cualquiera pueda continuar la tarea del compañero o detectar algún error tempranamente.

Se construyeron dos DMCs y un CDC para implementar el prototipo mencionado. Para la gestión y el acceso de consolas de texto, por las cuales los alumnos pueden programar el robot, se desarrolló un DMC en Python, que se ejecuta inmediatamente después de que los usuarios acceden al SR a través del protocolo SSH. Este DMC controla que el usuario que está accediendo tenga los permisos correctos en el laboratorio que quiere utilizar y que el laboratorio efectivamente esté ejecutándose. Si estas condiciones se cumplen, el DMC "conecta" al usuario al CDC correspondiente al recurso.

En la figura 4 se presenta una instancia completa de la arquitectura para el prototipo mencionado, a la izquierda de la nube se encuentran los clientes, basados en el Framework. Para los alumnos simplemente se necesitan terminales SSH para realizar la tarea de programación, mientras que el docente dispone también del front-end del ScreenController para poder controlar la interacción de los alumnos durante la práctica remota.

Del lado del servidor, a la derecha de la nube, los clientes se conectan todos al CDC asociado al recurso, y luego, cuando los alumnos finalizan la tarea asignada pueden "subir" el programa creado al Robot Físico a través de una torre infrarroja Lego que esta conectada al SR y tiene al Robot dentro de su alcance.

El otro DMC desarrollado, similar al sistema *WebCast* que ofrece OpenOffice [9] para exportar transparencias, sirve para que un docente pueda presentar contenidos teóricos a los participantes del laboratorio, como si fuera una presentación clásica de transparencias en clase. Este DMC se construyó usando el Servidor HTTP Apache [7], y el recurso en concreto es un programa CGI. Básicamente, en el programa CGI se mantiene el estado de la página que está viendo el "instructor", y los clientes (alumnos) mediante una consulta HTTP obtienen la URL de la siguiente transparencia. En el Cliente, se ejecuta un script que consulta de manera iterativa la dirección URL asociada al programa CGI para actualizar a la transparencia que el instructor desee. En este caso, el recurso es un *simple programa* que contiene un estado, que puede ser consultado por los clientes (Productor de información hacia el/los clientes) y puede ser modificado por el docente o instructor (Consumidor de información desde un cliente). Esta herramienta para la presentación de contenido teórico sincronizado, y de bajo consumo de recursos de red, junto con el uso de alguna herramienta de audioconferencia como Skype[8] resulta en una solución aceptable para el dictado de clases a distancia para 3 o 4 grupos de alumnos ubicados en distintas instituciones.

En el caso de que el uso de audioconferencia sea prohibitivo por el ancho de banda necesario, los alumnos disponen de una aplicación de Chat en Java, que ejecuta también el en SR.

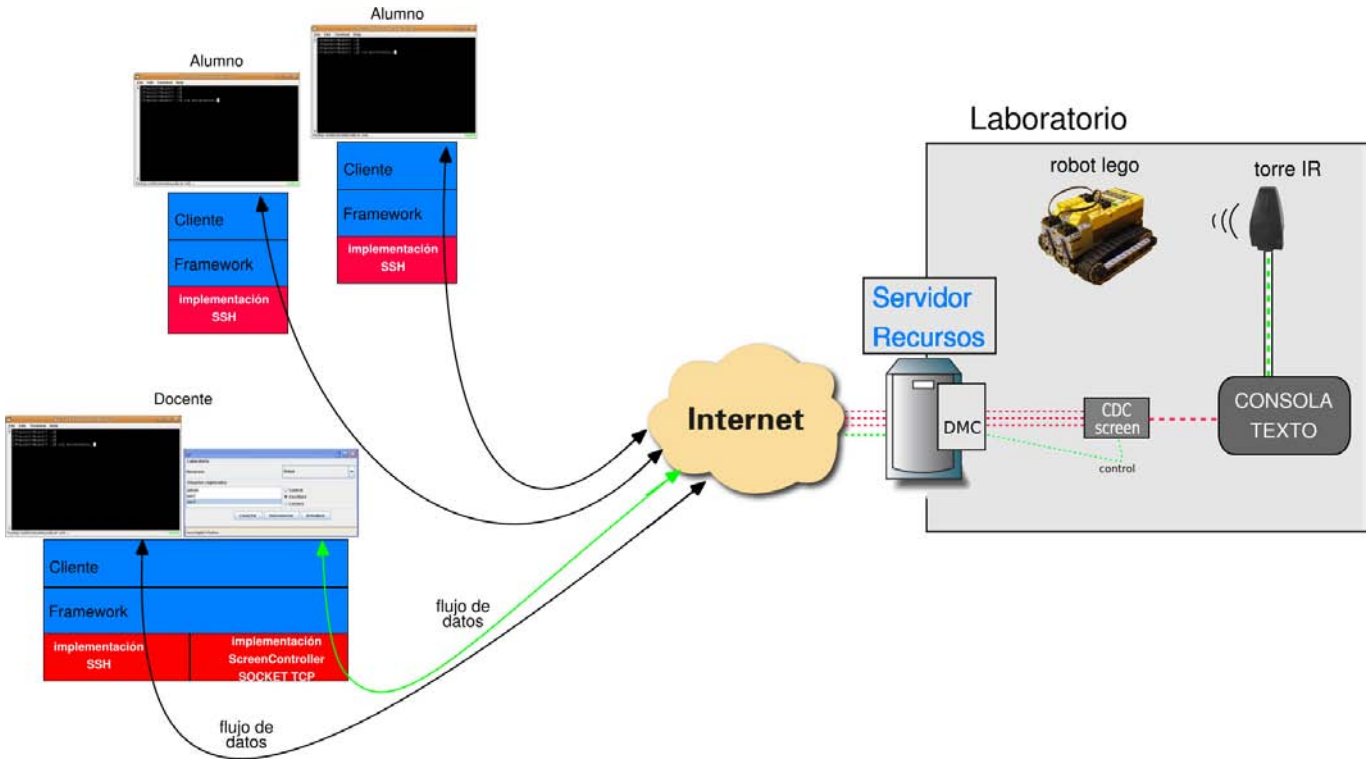


Figura 4: Instancia de la arquitectura para LAR de Programación

7. Conclusiones y trabajo futuro

El uso de un FOO para la construcción de aplicaciones que acceden a laboratorios físicos y virtuales de manera flexible e independiente, junto con la definición e implementación de entidades de control y mejora del acceso tradicional a los recursos en los servidores, permiten que la instauración de los laboratorios de acceso remoto para actividades de investigación sea cada vez más fácil de implementar. El desarrollo de nuevos prototipos de LARs como el expuesto, que logren un nivel de madurez en el software permitirá generar paquetes de software instalables, que además de estar disponibles para las instituciones también puedan ser utilizados por los usuarios finales de manera completamente local, simulando en su computador los recursos y evitando la necesidad de disponer de una conexión a Internet o de la presencia de otros alumnos para realizar una práctica. Mediante este uso, se complementa el concepto de LARs, con laboratorios virtuales y/o físicos pero utilizados localmente por el usuario final. Ejemplos de esta última modalidad de trabajo son la distribución de linux ADIOS[18] y el sistema VNUML[19]. ADIOS viene en un formato de livecd, y hace uso de máquinas virtuales UML y switches emulados por software para construir una topología de red para que un usuario pueda hacer prácticas de configuración de red en Linux. Por su parte, VNUML, utiliza también UMLs como emulación de computadores, pero además utiliza el simulador Dynamips[20] para simular completamente el hardware de los routers Cisco, permitiendo realizar complejos escenarios de networking con dispositivos de un lider mundial en estas tecnologías.

Referencias

[1] M. Bertogna, E. Grosclaude, R. del Castillo, F. Lopez Luro, C. Zanellato. “Arquitectura para Laboratorios Remotos Físicos y Virtuales”. CACIC 2005. pp 317-328.

[2] Grosclaude Eduardo, Bertogna Leandro, Lopez Luro Francisco, Zanellato Claudio, Sánchez Laura, Rodríguez Jorge, Del Castillo Rodolfo. “Experiencia con Laboratorio Remoto Colaborativo”. Exposición y Demostración. La Plata 2006

[3] Lopez Luro Francisco. “Framework basado en Java para el acceso Laboratorios Remotos Físicos y Virtuales”. Departamento de Ciencias de la Computación, Facultad de Economía y Administración. Universidad Nacional del Comahue, Diciembre 2006.

[4] GNU Screen, http://www.gnu.org/software/screen

[5] Java, http://java.sun.com/

[6] Java Applet, http://java.sun.com/applets/

[7] The Apache HTTP Server Project, http://httpd.apache.org/

[8] Skype Internet Phone, http://www.skype.com

[9] OpenOffice.org, http://www.openoffice.org/

[10] Report of the Expert Meeting on Virtual Laboratories - UNESCO, 2000, Paris.

[11] Bjedov, G. Utilizing the World Wide Web and the Internet to Facilitate Learning in Large Classes. Frontiers in Education, Conference Session 2c5, 2c5.13, 1995.

[12] Ralph E. Johnson. Frameworks = (Components + Patterns). How Frameworks compare to other object-oriented reuse techniques. Communications of the ACM. October 1997/ Vol. 40.,No 10.

[13] Ralph E. Johnson. Components, Frameworks, Patterns. Department of Computer Science, University of Illinois. 1997.

[14] Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

[15] Mark Grand. Patterns in Java: a catalog of reusable design patterns illustrated with UML, volume 1. Wiley Computer Publishing, 1998. ISBN 0471258393

[16] Patrones de Diseño, http://en.wikipedia.org/wiki/Design_pattern_(computer_science)

[17] Arquitectura Cliente-Servidor. http://en.wikipedia.org/wiki/Client-server.

[18] ADIOS Linux. http://os.cqu.edu.au/adios/index.html

[19] VNUML. http://www.dit.upm.es/vnumlwiki/index.php/Main_Page

[20] Dynamips. http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator

WORKSHOP DE INGENIERÍA INFORMÁTICA APLICADA A LA EDUCACIÓN

Modelo de Gestión del Conocimiento para Departamentos Académicos

Gregorio N. Tkachuk
Universidad Nacional de Santiago del Estero, Facultad de Ciencias Exactas y Tecnologías,
Santiago del Estero, 4200, República Argentina
gregorio@unse.edu.ar

Abstract

Knowledge Management involves a set of processes that deals with the knowledge of the organizations. It increases organizational creativity and guarantees organizational success because it is based in the most valuable resource the institutions have nowadays: knowledge or intellectual capital.

Although several models of Knowledge Management for enterprise context are available -that measures and optimizes human and technological capital, there are few models for non profit organizations. Moreover, there is no available model for educational context to improve organizational effectiveness and efficiency.

The main objective of this paper is to propose a Knowledge Management Model for Academic Departments. This work is a partial result of a case study research developed at the Informatics Department of the National University of Santiago del Estero. Previously, the Knowledge Domain Map is presented, which was carried out using an own symbolic language. Then, the new model is proposed, based on the 'virtual communities' model, on the Intellectus model and on the Knowledge Domain Map.

Keywords: Knowledge Management, Educational Context, Knowledge Map, Knowledge Management Model.

Resumen

La Gestión del Conocimiento abarca un conjunto de procesos que administra el conocimiento de las organizaciones. Logra aumentar la creatividad y garantizar el éxito organizacional, ya que se fundamenta en el recurso más valioso con que actualmente cuentan las instituciones: el conocimiento o capital intelectual.

Existen diversos modelos de Gestión del Conocimiento para el ámbito empresarial que indican cómo medir y optimizar el capital humano y el capital tecnológico de una organización. Sin embargo, en las organizaciones sin fines de lucro los modelos son escasos. Más aún, en el contexto educativo, no se dispone de modelos de Gestión de Conocimiento que mejoren la eficacia y la eficiencia de las instituciones.

El objetivo de este trabajo consiste en proponer un Modelo de Gestión de Conocimiento para Departamentos Académicos, tomando como estudio de caso el Departamento de Informática de la Universidad Nacional de Santiago del Estero. Con tal fin, se presenta el Mapa de Conocimientos del dominio, elaborado con un lenguaje simbólico propio. Luego, se presenta el modelo construido tomando como base el modelo de comunidades virtuales, el modelo Intellectus y los conocimientos del dominio.

Palabras claves: Gestión del Conocimiento, Contextos Educativos, Mapa de Conocimiento, Modelo de Gestión del Conocimiento.

1 Introducción

Las universidades argentinas son organizaciones educativas complejas que actualmente están sumergidas en una profunda crisis institucional y, por lo tanto, necesitan ser gestionadas sistémicamente para elevar su excelencia académica en el contexto mundial [6]. El conocimiento constituye el principal flujo de estas organizaciones. Por ello, se sostiene que la universidad argentina es una organización que necesita gestionar eficaz y eficientemente su conocimiento. Y la Gestión del Conocimiento (GC) constituye el instrumento óptimo para lograr para ser instituciones académicamente exitosas [5].

La GC es una estrategia surgida recientemente en la Administración que optimiza la competitividad de las empresas. Consiste en un conjunto de procesos que facilitan y optimizan la captación, descubrimiento, almacenamiento, difusión y transmisión del conocimiento [7]. Para lograr esto, la GC se apoya en el estímulo y protección del conocimiento de los recursos humanos –lo cual permite crear nuevos conocimientos- y en el uso de las tecnologías de la información –lo cual permite difundir y compartir el conocimiento para transformarlo en conocimiento organizacional. La GC aborda, entre otros temas:

- La captura de Conocimiento, es decir, la creación de documentos, su conversión, su revisión, etc.
- El desarrollo de infraestructuras tecnológicas y aplicaciones para la distribución del conocimiento.
- La formación de los empleados para la creación de conocimiento.
- La mentalización de los empleados para que compartan su conocimiento.

¿Qué es el conocimiento? Los datos de una organización, representados en forma estructurada, una vez asociados a un objeto, se convierten en información. La información, asociada a un contexto y a una experiencia, se convierte en conocimiento. Existen dos tipos de conocimiento, el tácito y el explícito. El conocimiento tácito es aquel que no está escrito en ningún recurso de la empresa; es el que se encuentra en el pensamiento de cada persona que trabaja en la empresa, es el conocimiento que no está en la empresa cuando una persona se enferma o se retira a su casa. El conocimiento explícito es el que está plasmado en los manuales de normas y procedimientos, está en las bases de datos, en los sistemas de información de la empresa.

Los seres humanos están bastante desarrollados en lo que se refiere a ciertas habilidades del conocimiento: para entender el conocimiento, interpretarlo dentro de un contexto más amplio, combinarlo con otros tipos de información, o sintetizar varios significados en forma de conocimiento.

Por otra parte, computadoras y sistemas de comunicaciones son excelentes en la captura, transformación, y distribución del conocimiento y, en esta actividad, son más capaces que las personas.

Dado esta mezcla de habilidades, es necesario construir ambientes de gestión del conocimiento híbridos, en los que intervienen humanos y tecnología de manera complementaria. Se deben construir fábricas de conocimiento combinando personas y sistemas informáticos [1].

Existen diversos modelos de GC que se aplican a contextos empresariales (donde surgió la GC) que miden, cuantifican y optimizan la GC en función de los recursos intangibles (capital humano, capital intelectual y capital tecnológico) de la organización. Generalmente el resultado de la implementación de la GC se mide en virtud del valor agregado de la organización, es decir, la diferencia entre el valor de bolsa y el valor registrado en los libros contables. Sin embargo no

existen modelos de GC que se apliquen a entornos educativos en base a los cuales se pueda implementar Sistemas de GC en unidades u organizaciones académicas.

Esta publicación consiste el resultado parcial de una investigación más amplia que se lleva a cabo en el Grupo de GC de la Universidad Nacional de Santiago del Estero. Uno de los objetivos de la misma consiste en desarrollar e implementar Sistemas de GC en Departamentos Académicos. Para lo cual es necesario, previamente, elaborar modelos de GC para contextos educativos.

El trabajo sigue la siguiente estructura: en principio se abordan los modelos de GC que servirán de base para el modelo propuesto; luego, se presenta el Mapa de Conocimiento elaborado para el caso, detallando el proceso seguido y el lenguaje usado para su construcción; posteriormente, se presenta el Modelo de GC para Departamentos Académicos; finalmente, en las conclusiones se hace referencia a la implementación de este modelo en un Sistema de GC.

2 Modelos de GC

No existen en la actualidad modelos claros para una gestión del conocimiento uniforme, si bien es necesario reconocer la importancia de diversas variables en dicho proceso de gestión. Este proyecto de elaboración de un modelo para entidades académicas universitarias se basa en dos modelos: el modelo de comunidades virtuales (basados en agentes de conocimiento) y un modelo para entidades educativas propuesto por el autor en base al modelo Intellectus y su adaptación a entidades sin fines de lucro.

2.1 Modelos basados en Agentes de Conocimiento

Los agentes de conocimiento forman parte de un sistema en el que los mismos interactúan creando su vez nuevas relaciones causales y consecuenciales. Son los siguientes: personas, equipos y organizaciones [4].

Las personas son la raíz de todo conocimiento. La ciencia, desde su origen, busca conocer el funcionamiento de la naturaleza y de la especie humana.

Los equipos constituyen la unidad básica de la estructura del conocimiento. Cada equipo tiene sus propios individuos, normas, objetivos y responsabilidades.

Las organizaciones son sistemas complejos que tratan de aportar un valor añadido a la sociedad en la que se hallan. Por ejemplo: empresas, centros de estudio, administraciones públicas, etc.

Las continuas interacciones entre los tres agentes de conocimiento provocan una relación dinámica y no lineal. Según E. Bueno [2], estas relaciones marcan cuatro dimensiones que toma el conocimiento cuando se pone en interacción con estos tres agentes. Estas dimensiones son: ontológica, epistemológica, estratégica y sistémica.

Dimensión ontológica:

- Solo los individuos crean conocimiento
- Dichos conocimientos pueden ponerse a disposición de la organización
- El conocimiento de las organizaciones es más que la suma de los conocimientos de los individuos que la componen

Dimensión epistemológica:

- El conocimiento explícito es fácil de articular y verbalizar; es sistémico y objetivo; racional y lógico; digital, secuencial, referido al pasado y libre del contexto en el que se produce
- El conocimiento es difícil de articular y verbalizar; subjetivo, arraigado en la experiencia y las emociones; analógico, simultáneo, sobre el presente y dependiente de un contexto

Dimensión estratégica:

- El conocimiento aparece como un recurso de primer orden, ya sea explícito o tácito
- El directivo puede gestionar conocimiento explícito (base de dato, verbigracia), no se puede gestionar conocimiento tácito (relaciones, experiencias)
- El directivo, por tanto, debe crear contextos y provocar interacciones

Dimensión sistémica:

- Es necesario especificar el alcance de la observación
- Cada vez hay menos claridad en la frontera entre lo interno y lo externo

Las comunidades virtuales garantizan una permanente comunicación entre los agentes de conocimiento. Brindan la oportunidad de definir la organización y comunicárselo a todos los integrantes de la misma.

2.2 Modelo Intellectus para Organizaciones sin Fines de Lucro

Actualmente existen diversos modelos de GC para empresas en general. Sin embargo, no es apropiada la aplicación de éstos a las entidades educativas. Esto hace que sea necesario el diseño y la elaboración de modelos adaptados que respondan a sus necesidades específicas.

Este trabajo se basa en la adaptación del modelo Intellectus[1] para organizaciones sin fines de lucro propuesto por Gallego y Ongallo [4]. Este modelo adaptado, parte de los elementos de capital intelectual establecidos con carácter general. Luego define variables específicas que permiten establecer información de los activos intangibles característicos del sector sin fines de lucro. A partir de este modelo de Gallego y Ongallo, se define un modelo específico para instituciones educativas.

En la tabla 1 se presentan los elementos intangibles que aportan información relevante sobre el capital intelectual de las organizaciones educativas, especificando variables para cada uno de ellos. Estos elementos y variables constituyen la estructura básica de un modelo de capital intelectual para este tipo de organizaciones, deben ser completados con el establecimiento de indicadores de medición.

3 Mapa de Conocimiento del Departamento Académico

Este proyecto se lleva a cabo en el Departamento Académico de Informática de la Facultad de Ciencias Exactas y Tecnologías de la Universidad Nacional de Santiago del Estero. A partir de este estudio de caso se pretende obtener un Sistema de Gestión de Conocimiento para Departamentos Académicos Universitarios que pueda ser implementado en otras unidades académicas.

[1] El modelo Intellectus fue elaborado en el Instituto de Administración de Empresas de la Universidad Autónoma de Madrid, 2003.

Tabla 1: Elementos y variables del modelo de GC Intellectus, adaptado a organizaciones educativas.

Capital humano	**Variables**
Valores y actitudes (ser + estar)	Voluntariedad Solidaridad Flexibilidad Creatividad
Aptitudes	Educación reglada Formación especializada Conocimiento del destinatario de los servicios
Capacidades (saber hacer)	Aprendizaje Trabajo en equipo Liderazgo
Capital Organizativo	**Variables**
Cultura	Visión Valores culturales Clima social
Estructura	Autonomía organizativa Dinamismo organizativo
Aprendizaje organizativo	Entornos de aprendizaje Pautas organizativas Captación y transmisión de conocimiento Creación y desarrollo de conocimiento
Procesos	Dirigidos al cliente interno (docentes, alumnos, profesionales) Dirigidos a las autoridades (Secretarios, decano, rector)
Capital tecnológico	**Variables**
Dotación tecnológica	Esfuerzo en la adquisición de tecnología Dotación de tecnologías de la información y de las comunicaciones
Innovación	Innovación de proceso Innovación de gestión Innovación social

3.1 Los Procesos de la Gestión del Conocimiento

Los procesos inherentes a la gestión del conocimiento, según Valhondo [7], son aquellos que permiten descubrir, crear, capturar, clasificar, compartir y difundir el conocimiento (ver figura 1). Teniendo en cuenta esto, los procesos que se identificaron en el Departamento de Informática como procesos de conocimiento son los siguientes:

1. Identificación del conocimiento deseado
2. Identificación del conocimiento disponible
3. Generación de nuevos conocimientos
4. Representación del conocimiento disponible
5. Adquisición del conocimiento externo
6. Clasificación, estructuración y almacenamiento del conocimiento
7. Transmisión y difusión de los conocimientos
8. Captura del conocimiento
9. Asimilación del conocimiento (aprendizaje)

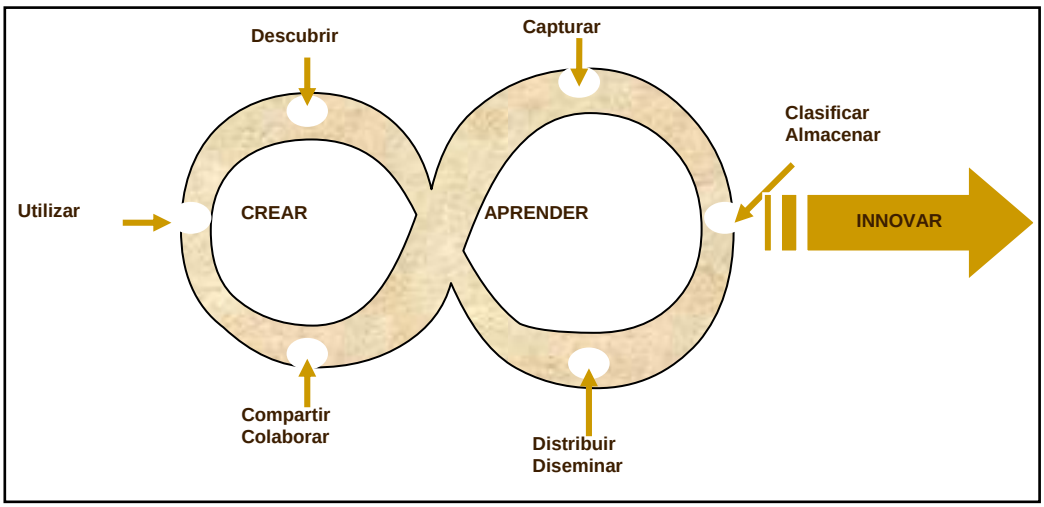


Figura 1: Procesos de la Gestión del Conocimiento según Valhondo.

A su vez, estos procesos están relacionados con los diversos conocimientos identificados en el Departamento de Informática. Estos conocimientos se presentan a través de un Mapa de Conocimiento.

3.2 Mapas de Conocimiento

Un mapa del conocimiento se basa en la identificación de requerimientos de conocimiento de todos los procesos que tienen una fuerte dependencia de los activos intelectuales. Se caracteriza por los atributos siguientes:

- Constituye la recopilación de los conocimientos disponibles en una unidad u organización.
- Distingue los conocimientos explicitados y documentados, del conocimiento tácito que tienen las personas relevantes.
- Prioriza y agrupa el conocimiento.
- El mapa indica, además, cómo llegar a este conocimiento relevante: qué personas lo tienen, en qué soporte se encuentra, etc.
- Permite identificar las lagunas de conocimiento.

Un mapa del conocimiento puede ser utilizado como herramienta de diseño y mantenimiento de un sistema de gestión del conocimiento. Va más allá de la aplicación de una taxonomía para la navegación en las fuentes del conocimiento de la empresa. Debe ser una herramienta útil para la definición de la estrategia de la organización.

3.3 Mapas de Conocimiento del Departamento Académico

En la figura 2 se presenta el Mapa de Conocimientos del Departamento de Informática, en el cual se muestran los conocimientos, tipos de conocimientos, procesos con los cuales están involucrados, las relaciones entre ellos y con el capital humano. A continuación se describe el procedimiento con el cual se elaboró el Mapa.

Primero se identificaron los principales procesos de la organización, en virtud de las herramientas de captura de conocimiento de los integrantes de la organización. Si bien, existen diversos procesos, los seleccionados para elaborar el modelo son aquellos en los cuales se detectaron la mayor cantidad de relaciones entre los conocimientos. Estos son:

- P1: Definición de Equipos Cátedras
- P2: Llamados a Concursos Docentes
- P3: Asignación de Turnos del Laboratorio Informático a las Asignaturas
- P4: Gestión de los Recursos Tecnológicos del Departamento

Una vez definidos los procesos, se identificaron los conocimientos que corresponden a cada uno de ellos. A continuación se presenta una lista de los principales conocimientos involucrados en la gestión departamental académica de Informática:

- Docentes: Nombre, Mail, Interno, Cargo, Títulos de grado y postgrado, Dedicación, Asignaturas en las que trabaja, Áreas de investigación, Dominios de conocimiento, Desempeño, Declaración jurada
- Asignaturas: Nombre, Modulo y año, Carrera a la que pertenece, Correlativas posteriores y anteriores, Carga horaria, semanal y total, Contenidos mínimos, Objetivos, Planificación, Horarios aulas, Equipo cátedra, Exámenes.
- Concursos: Asignatura, Cargo, dedicación y condición ordinario o interino, Fecha, Tribunal, Ganador.
- Pasantias: Lugar o empresa, Perfil requerido, Alumnos (Nombre), Periodo, Tutor, Resolución, Resultados
- Reglamentaciones: Resolución, Tema, Síntesis
- Recursos: Tipo (didáctico o informático), Nombre, Descripción, Estado, Disponibilidad, Reglamentación de uso, Nro de inventario
- Laboratorios: Nombre, Responsable, Equipo de trabajo, Recursos, Cantidad, Software disponible, Reglamento de uso, Reservas, Disponibilidad, Horarios de atención
- Biblioteca: Nombre, Bibliografía disponible, Reglamento de uso, Responsables (Nombre, Mail), Horarios de atención
- Procesos: Gestión de presupuesto, Solicitud de ayudantes de 2, Solicitud de renovación de interinos, Solicitud de nuevos cargos, Solicitud de nuevo equipamiento, Gestión de tramites docentes, Asignación de tribunal para trabajo final, Solicitud de bibliografía y software, Armar equipo cátedra, Comunicación de novedades a docentes

Luego se clasificaron los conocimientos conforme a su tipo: explícitos y tácitos. Además, para los explícitos se identificó el tipo de soporte en el cual está documentado o almacenado, así como también su disponibilidad actual para las personas del departamento, el lugar donde están alojados. Para los conocimientos tácitos se identificó quiénes son las personas que lo poseen; esto es importante porque el sistema de GC debe tratar de producir conocimiento explícito a partir del implícito.

Finalmente, se identificaron las relaciones entre los diversos conocimientos. Se puso hincapié en las más relevantes y trascendentes para evitar aumentar la complejidad del mapa de manera innecesaria.

El mapa de la figura 2 fue construido en base a un lenguaje gráfico ideado específicamente para esta finalidad, sus símbolos se muestran en la Tabla 2.

Tabla 2: Lenguaje gráfico usado para la construcción del Mapa de Conocimiento.

Símbolo	Representación
	Conocimiento Explícito.
	Conocimiento Tácito.
	Persona poseedora de un conocimiento tácito específico (capital humano). Su rol en la organización.
	Conocimiento explícito almacenado en documentos no digitalizados.
	Conocimiento explícito almacenado en soporte digital. Pueden estar alojados en diversos medios: discos rígidos de puestos de trabajo, discos de servidores, disquetes. También pueden estar en distintos formatos: pdf, doc, bmp, etc.
	Relación entre Procesos.
	Relación entre Conocimientos.
	Deriva cada conocimiento (ya sea tácito o explícito) hacia el soporte en el que está almacenado.

El Mapa de Conocimiento del Departamento de Informática muestra cómo se distribuyen los conocimientos en la actualidad en el departamento, mostrando el conocimiento que actualmente posee y los lugares donde actualmente están alojados y que puede tener acceso el departamento. En la figura 2 solo se muestra un estrato del mapa total: el proceso P1 y el proceso P3.

5 Modelo de GC para Departamentos Académicos

El modelo elaborado se basa en todos los conceptos que se abordaron hasta este punto. Es importante realizar una descripción del modelo considerando esos fundamentos. Por ello, a continuación se presenta una caracterización del Modelo de GC para Departamentos Académicos organizada en tres aspectos: dimensiones, arquitectura, conocimientos.

a) Dimensiones. El modelo propuesto fue elaborado considerando las dimensiones identificadas en las comunidades virtuales donde los que interactúan los agentes de conocimiento. Estas dimensiones son: Dimensión ontológica (qué constituye conocimiento y quién lo posee), Dimensión epistemológica (diferenciar los conocimientos explícitos y tácitos), Dimensión estratégica (los directivos necesitan acceder al conocimiento), Dimensión sistémica (delimitar lo interno de lo externo).

b) Arquitectura. La arquitectura del modelo fue ideada en base al modelo Intellectus adaptado para organizaciones educativos, en el cual se identificó como principales componentes: Conocimiento

Humano, Conocimiento Tecnológico y Conocimiento Organizacional. En la figura 3 estos componentes están representados por grandes círculos. En vistas al desarrollo e implantación del Sistema de GC, el más importante es el conocimiento tecnológico o arquitectura tecnológica de conocimiento, que se encuentra detallada en la figura y se explicará más adelante.

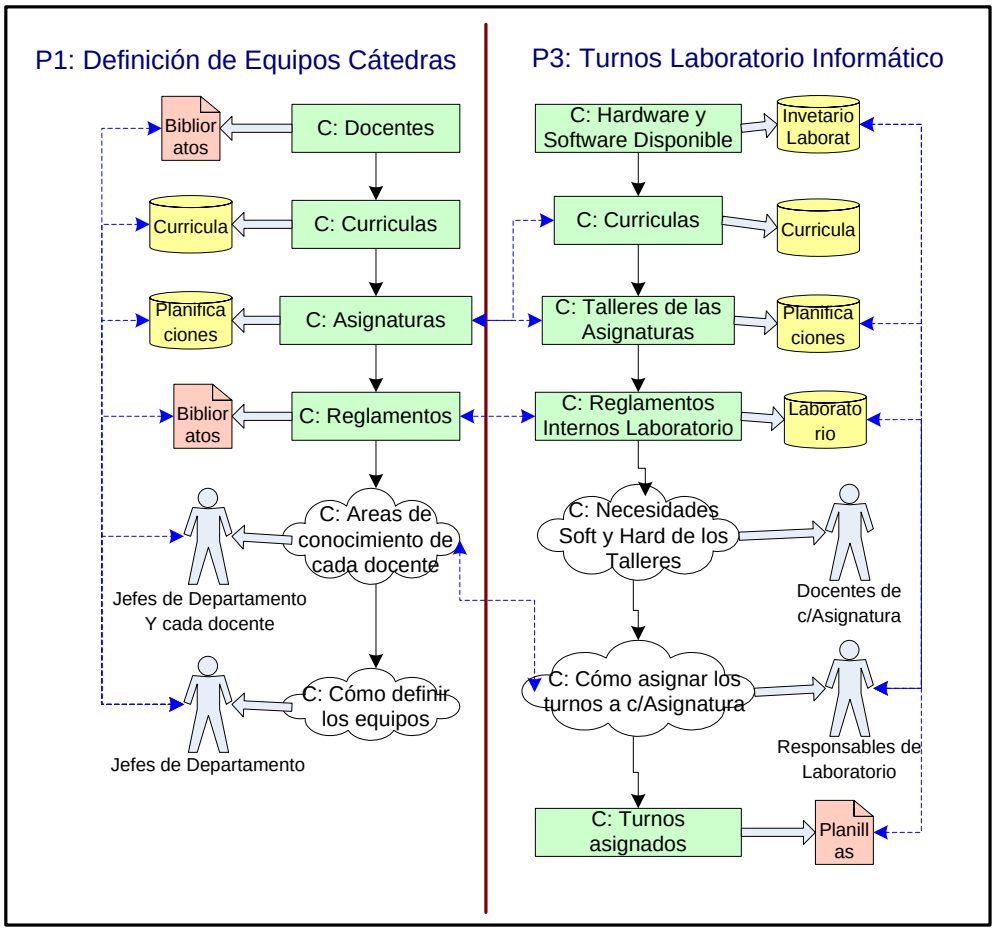


Figura 2: Mapa de Conocimiento del Departamento de Informática. Vista parcial: proceso P1 y proceso P3.

c) Conocimientos. El modelo se basa en los conocimientos y relaciones identificados en el Mapa de Conocimientos presentado en el punto anterior. En este punto es importante resaltar que el Mapa de Conocimientos indica los actuales lugares de almacenamiento del conocimiento (y su soporte actual); mientras que en el modelo de GC se indican los lugares de almacenamiento óptimos que el Sistema de GC considera para cada conocimiento.

En la figura 3 se muestra el Modelo de GC. Si bien no están graficadas las dimensiones en forma explícita, están implícitamente consideradas. En cuanto a la dimensión ontológica, no se especifican cada conocimiento puesto que eso se encuentra en el Mapa de Conocimiento; considerando la dimensión ontológica, se diferencian los conocimientos tácitos y explícitos utilizando los mismos símbolos del Mapa de Conocimiento (el conocimiento humano está constituido por conocimiento tácito mientas que en la capa de almacenamiento se encuentran los conocimientos explícitos); la dimensión estratégica está implícita en la intención de desarrollar la aplicación que permita automáticamente acceder a todos los conocimientos; y considerando la dimensión sistémica, se delimitaron los conocimientos y ámbitos específicos de la unidad bajo análisis.

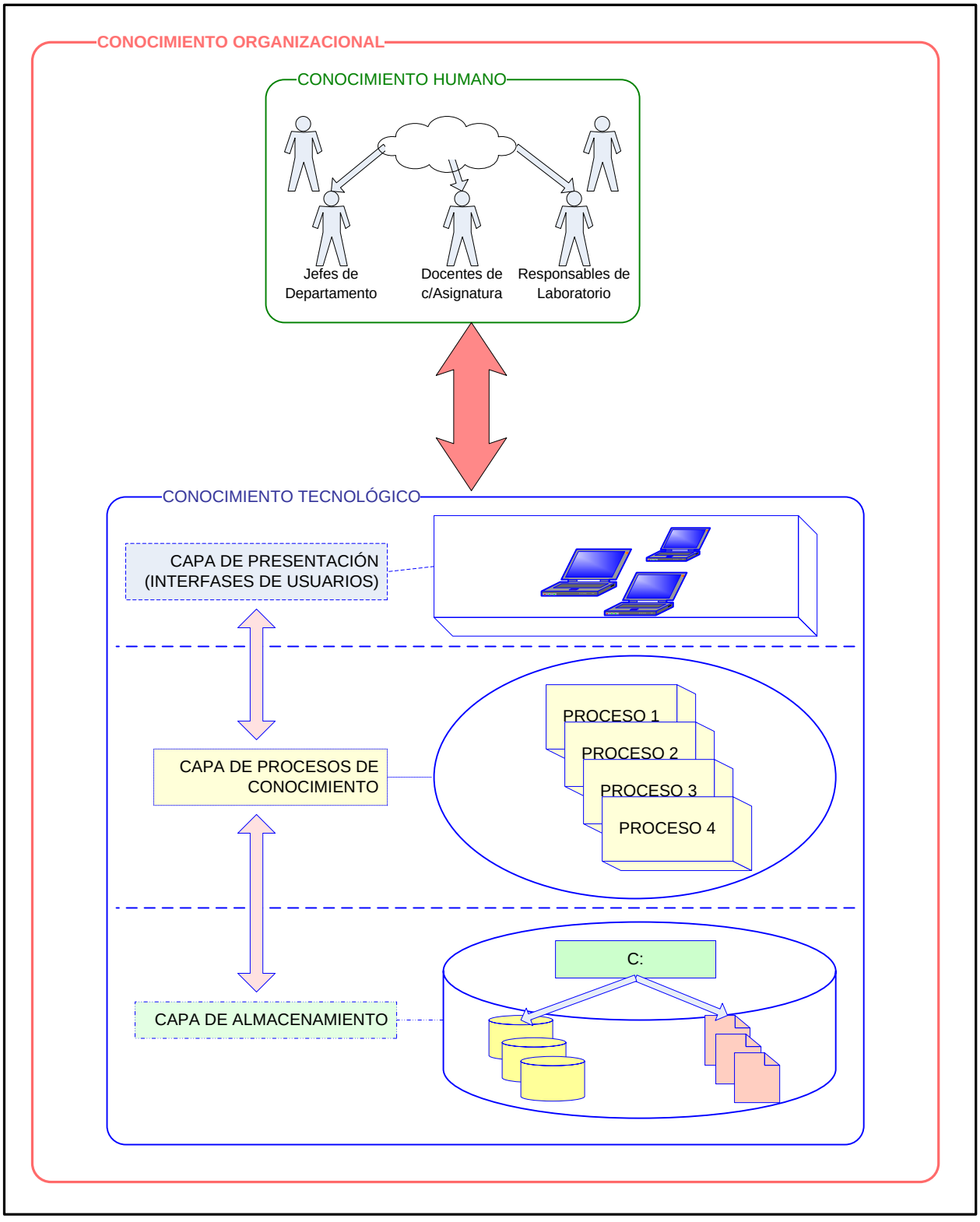


Figura 3: Modelo de Gestión Académica para Departamentos Académicos.

El aspecto que principalmente muestra el modelo es la arquitectura. Sobre la base del modelo Intellectus adaptado para entidades educativas, presenta tres componentes arquitectónicos: el

conocimiento humano (conocimiento tácito que reside en la mente de las personas que desempeñan los roles en el Dpto. de Informática), es importante porque es el único capaz de generar nuevos conocimientos y aumentar la creatividad, en base al uso de las tecnologías de conocimiento; el conocimiento tecnológico, que constituye la arquitectura tecnológica; y el conocimiento organizacional que es el más importante ya que es el conocimiento que se desea lograr y que surge de la simbiosis e interacción entre el conocimiento humano y el conocimiento tecnológico.

En cuanto a la arquitectura tecnológica, está diseñada en tres capas: de presentación, de procesos de gestión del conocimiento y de almacenamiento. La primera hace referencia a la interfaz de usuario; la segunda hace referencia a los procesos que debería tener el Sistema de GC para permitir que se den los procesos de GC (almacenar, capturar, diseminar, compartir, etc.). La capa de almacenamiento en aquella en la cual se alojarán todos los conocimientos explícitos. El modelo propone que incluya tanto los conocimientos digitales como los documentados sin digitalizar (es decir, el Sistema de GC deberá tener digitalizados todo el conocimiento explícito).

Tal cual sostiene el modelo Intellectus, este modelo debe permitir la distribución y compartir el conocimiento entre todo el capital humano, de forma tal que se pueda construir a partir de él un sistema que propicie la cultura de compartir el conocimiento, lo cual a su vez permite externalizar los conocimientos tácitos.

6 Conclusión y Trabajos Futuros

El modelo presentado en este artículo se propone como alternativa viable para incorporar la Gestión del Conocimiento en la gestión universitaria, ya que el conocimiento es el producto original y legítimo de las universidades. Esto aumentaría la eficacia y eficiencia de la gestión universitaria.

En trabajos futuros se abordará la instancia posterior dentro de la metodología de desarrollo e implementación de un Sistema de GC, que consisten en la construcción del Sistema Software (o sistema tecnológico) que permita implementar el modelo elaborado en una aplicación concreta que facilite los procesos de conocimiento (capturar, almacenar, distribuir, difundir, etc.).

En esta instancia, de acuerdo al trabajo realizado sobre los procesos y los conocimientos del dominio, se puede anticipar que el sistema de GC a desarrollar tendrá las siguientes características:

- Propiciará la cultura de compartir el conocimiento académico
- Se basará en un portal que contenga accesos a los diversos tipos de conocimientos que posee el Departamento Académico
- Permitirá el acceso al conocimiento desde cualquier lugar y en cualquier momento
- Optimizará la comunicación entre los recursos humanos poseedores del capital intelectual
- Incluirá un centro de documentación centralizado que permita alojar archivos digitalizados en cualquier formato
- Involucrará una base de datos, centralizada o distribuida, que garantice la integridad de los conocimientos y facilitará la localización de los mismos
- Contribuirá a la creación de nuevos conocimientos y a la optimización de decisiones estratégicas
- Contendrá un sistema de medición del capital intelectual de una organización educativa

El Sistema de GC que se construirá y los resultados de su implantación serán objeto de publicaciones que el grupo de investigación emitirá en un futuro a corto plazo.

Referencias

[1] Barnes Stuart. Sistemas de Gestión del Conocimiento. Teoría y Práctica. Ed. Thomson, Madrid, 2002. ISBN 84-9732-164-2.

[2] Bueno, E. *Perspectivas sobre dirección del conocimiento y capital intelectual*. Madrid, Instituto Universitario Euróforum Escorial, 2000.

[3] Fundación Iberoamericana del Conocimiento. *GestiónDelConocimiento*, España, disponible en url: <www.gestiondelconocimiento.com>, fecha de acceso: 01/05/06.

[4] Gallego, J. Ongallo, C., De Tena, R. *Conocimiento y Gestión*. Pearson Educación, S.A. Madrid, 2004.

[5] Herrera, S., Tkachuk G., Campos M. Gestión del Conocimiento Universitario. Una Propuesta Sistémica para Elevar la Excelencia Académica en las Universidades Argentinas. *1er. Congreso Regional de la Asociación Latinoamericana de Sistemas (ALAS)*, Buenos Aires, Agosto 2006.

[6] Touraine, A. El país paga caro la falta de gestión. *La Nación, Cultura*. Buenos Aires, Febrero 2006. Disponible en url: www.lanacion.com.ar/784235. Fecha de acceso: 01/05/06.

[7] Valhondo, D. *Gestión del Conocimiento. Del mito a la realidad*. Ed. Díaz de Santos, Madrid, 2003. ISBN 84-7978-542-X.

MODULE FOR ADAPTIVE TEACHING PLANNING FOR A TUTORING SYSTEM

Jorge Omar Ceyca Ceyca. Rodolfo A. Pazos Rangel,
Jorge Alberto Ruiz Vanote
Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET),
Interior Internado de Palmira s/n, Palmira, Cuernavaca, Morelos, México,
C.P. 62490, Tel. (777) 318-77-41,
E-mail: jceyca@gmail.com

and

Ocotlan Díaz Parra
Universidad Autónoma del Estado de Morelos (UAEM),
Av. Universidad 1001. Col. Chamilpa, Cuernavaca, Morelos, México,
C. P. 62209, Tel. (777) 329-70-00,
E-mail: jceyca@gmail.com

Abstract

The teaching process has been strengthened by the emergence of modern computer technologies, which have permitted developing distance education mainly through the use of the Internet. In this field tutoring systems have become one the most benefited and important tools for teaching. However, adapting tutoring systems to students' capacity is one the issues that has just begun being explored [1, 2]. This paper describes a module for adaptive teaching planning for a tutoring system that is under development [3, 4]. This module modifies the teaching contents depending on the estimation of two of students' cognitive abilities: memorization and comprehension.

Keywords: Adaptive Teaching Planning, Tutoring System, and Cognitive Abilities.

Abstract

La enseñanza ha sido fortalecida con la aparición de las modernas tecnologías computacionales, que han permitido desarrollar la educación a distancia principalmente con el uso de la Internet, siendo los sistemas tutores una de las herramienta más beneficiadas e importantes para esté fin. Sin embargo, la adaptación de los sistemas tutores a las capacidades de los estudiantes es un punto que a penas está siendo explorado [1, 2], es por ello que dada la experiencia en el CENIDET para desarrollar sistemas tutores [3, 4] se implementó un módulo de planificación adaptativa que modifica el contenido de enseñanza dependiendo de la cuantificación de la calidad de las habilidades cognoscitivas de memorización y comprensión del estudiante.

Palabras claves: Planificación Adaptativa de Enseñanza, Sistema Tutor, y Habilidades Cognitivas.

1 INTRODUCTION

The teaching process requires a constant improvement in the existing methods, tools and systems for this purpose, mainly concerning distance education. This is not entirely new: distance education based on traditional means has a long history. The novelty consists of requiring that teaching be assisted by modern computer techniques, multimedia and communications for offering technologically advanced environments that permit to carry out teaching attractively and efficiently.

One of the modern technologies is the world communications network, Internet, which has had and will continue to have a large impact on different activities and areas of human development. There will be no human activity, which, one way or another, will not be affected by this new revolution that has expanded the limits of all the disciplines. This includes distance teaching, specially tutoring systems, whose range of action is almost limitless; i.e., they have the ability to reach any place in the world that is connected to the Internet.

Thus, the proliferation of Web sites that offer tutoring systems on different fields of knowledge constitutes a clear example. Such tutoring systems extend from those that are just simple digital books to the most sophisticated, which use artificial intelligence techniques that offer advanced interaction means with the student.

2 TUTORING SYSTEMS

There exist many definitions and terms for referring to tutoring systems. In the Military Handbook 284 part 3 and in the Military Standard 1379D, the term used for denoting these systems is interactive courseware, which is defined as follows [5]: “Computer program for controlling the student’s instruction, which determines the time and order of teaching. The student progresses through the sequence of events making decisions and choices. The instruction proceeds according to the student’s responses.”

The objective of tutoring systems is reproducing the teacher’s activities in the teaching-learning process, both individually and collectively. Though, the first tutoring system of this type was developed over 30 years ago, researchers have not devised a system that satisfies all the requirements achieved by a human teacher.

2.1 Architecture of a Tutoring System

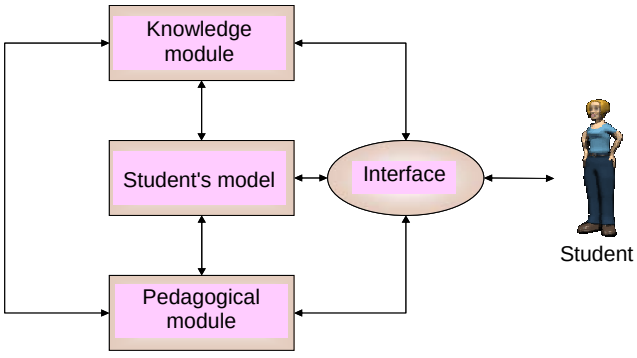


Figure 1: Natural division of an ITS.

A traditional way of representing tutoring systems consists of dividing its components in a natural way (Figure 1) [6, 7]:

a) Knowledge module.
b) Student's model.
c) Pedagogical module.
d) Interface.

The **knowledge module** stores the knowledge representation of the subject to be taught. It consists of the following basic elements: a knowledge base and an inference engine. The main methods for representing knowledge are: semantic networks or directed graphs, production systems or rules, agents, and case based reasoning.

The **student model** is a representation of the knowledge acquired by the student, as well as other characteristics among which the most relevant are cognitive abilities, attentiveness, dedication, number of reviews, etc. raditionally the techniques used for designing this model are overlay and buggy.

The **pedagogical module** can normally apply one or more teaching techniques and has the capacity of dividing the teaching plan into sub-goals for directing a student in a customized way. Additionally, it can dose up the contents according to the records of the general strategic decisions and the results of the student's historical behavior. The main didactical methods that have been used are passive student and participative student.

Finally, the **interface** processes the in and out communication flow. This is not an easy task in this kind of systems, since the communication channel between the student and the system is very narrow.

3 COGNITIVE ABILITIES

The word cognition refers to the abilities of the logical and rational thought of individuals [8]. Such abilities are responsible for facilitating the adequate integration of the individual with the surrounding environment. Each ability is constituted by cognitive abilities such as: memorization, comprehension, reasoning, analysis, etc.

In 1956 Benjamin Bloom leaded a group of educational psychologists that developed one of the most important classifications of intellectual behavior. This originated a taxonomy that includes three domains: cognitive, psychomotor and affective [9].

The cognitive domain (which is relevant for this investigation), is constituted by six levels (Figure 2) and is highly important, since it offers a starting point for the basic classification of the cognitive abilities required from the individuals for achieving the learning of knowledge.

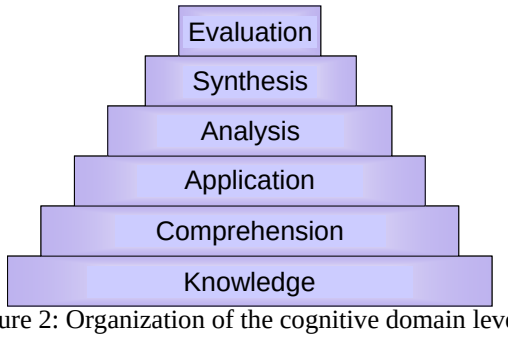


Figure 2: Organization of the cognitive domain levels.

Knowledge is defined as the act of remembering previously learned material. It presupposes remembering a rich gamut of materials that ranges from concrete facts to theories; but what is needed altogether is remembering the appropriate information.

Comprehension is defined as the capacity for acquiring the meaning of the material. It can be revealed when translating the material from one form to another (words to formulas), when interpreting the material (explaining it or summarizing it), and when estimating which will be the future trends (prediction of consequences or effects).

Application refers to the capacity of using the learned material in new and concrete situations. It can include the application of elements such as: rules, methods, concepts, principles, laws, and theories.

Analysis refers to the capacity of subdividing the material to be learned into its constituting parts, such that its organization structure can be understood. It includes the identification of the parts, the analysis of the relationships among parts, and the realization of the involved organization principles.

Synthesis implies joining parts, such that a whole is integrated, which may require the production of a unique communication (theme or discourse), an operation plan (investigation proposals), or a set of abstract relationships (plan for information classification).

Finally, **evaluation** consists of the capacity of judging the value of some material based on personal values and/or opinions, which results in a final product, with some given purpose, without real correct or incorrect answers.

4 MODULE FOR ADAPTIVE TEACHING PLANNING

The module for adaptive teaching planning is basically implemented above the pedagogical module according to the typical architecture of tutoring systems. The adaptive planning operates on the contents to be taught by the system and considers the memorization and comprehension cognitive abilities of the student. The specific consideration of the memorization and comprehension abilities is highly important in the students' learning process, since these are the first steps in the process (according to Bloom's taxonomy) through which higher levels of knowledge assimilation and processing are achieved.

For achieving adaptive teaching planning considering the aforementioned cognitive abilities, it is necessary to relate the knowledge about the course contents with the memorization and comprehension abilities required for the understanding of the material by the student.

4.1 Basic Functions of the Module

The adaptive planning module performs the following basic functions:

a) Measuring the student's memorization and comprehension abilities, in such a way that the student is not intimidated by this assessment.
b) Planning the contents of the topic to teach according to the assessment of the memorization ability.
c) Planning the contents of the topic to teach according to the evaluation of the comprehension ability.

d) Determining if low values of the estimated cognitive abilities are due to insufficient time devoted by the student to examine the material.

4.2 Database Schema of the Tutoring System

All the modules of the tutoring system are supported by a database constituted by 21 tables grouped into four classes (Figure 3):

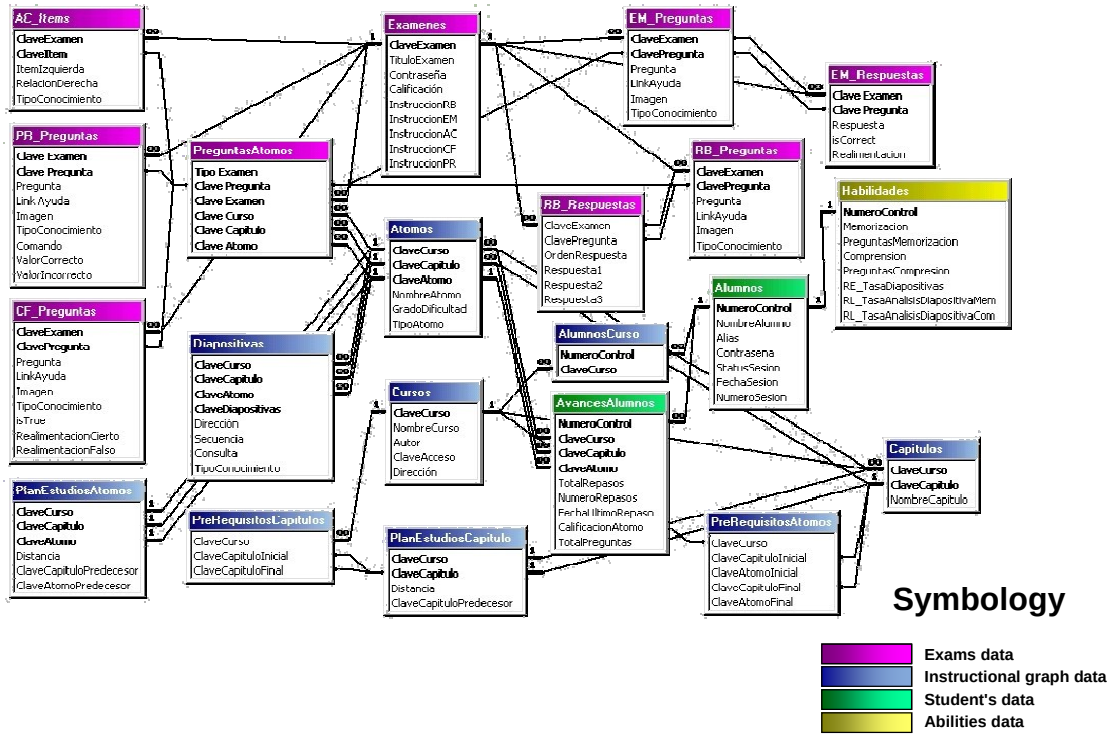


Figure 3: Database schema of the tutoring system.

a) Information of the course exams, which consider five different evaluation mechanisms.
a) General information on students (identification data).
b) Information on assessment of the students' cognitive abilities (memorization and comprehension).
c) Information of the student's model (representation of the students' learning degree).

4.3 General Architecture of the Tutoring System

The tutoring system used in this investigation is being developed at the *Centro Nacional de Investigación y Desarrollo Tecnológico* [3, 4]. The tutoring system modules have been adapted for the adequate operation of the adaptive teaching module, which has resulted in the current system architecture, shown in Figure 4.

As can be seen in Figure 4, the four modules in the lower part of the figure constitute the tutoring system proper; while the four upper modules are authoring tools for facilitating the creation of courses in different fields of knowledge.

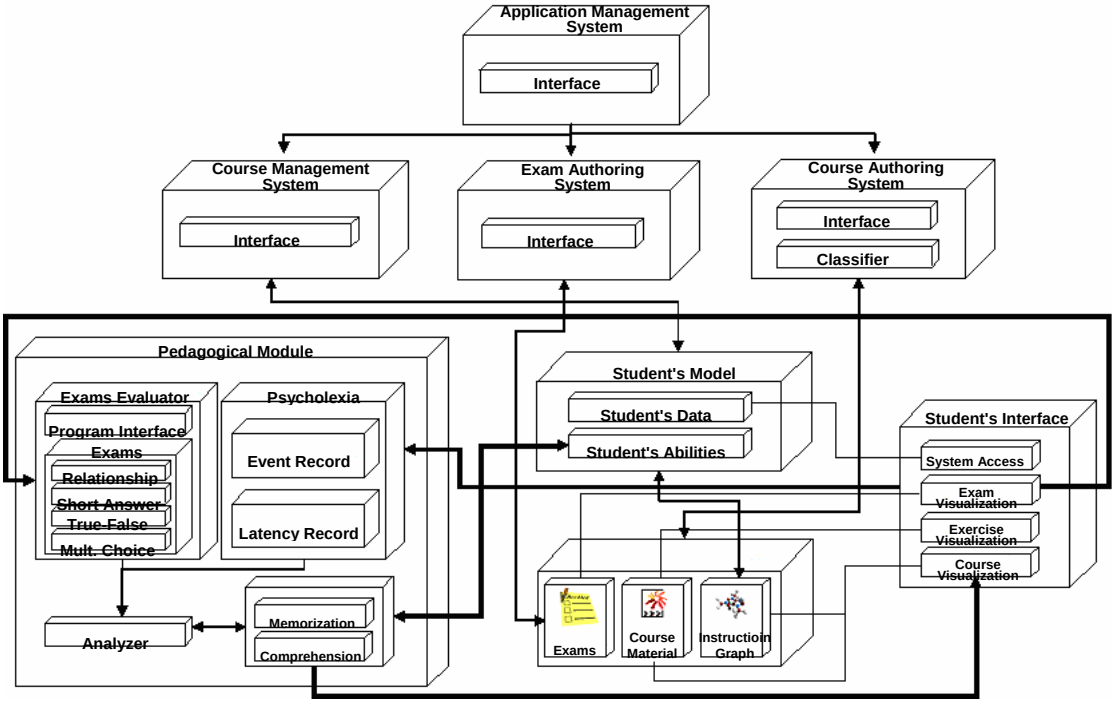


Figure 4. General architecture of the tutoring system.

4.4 Architecture of the Adaptive Planning Module

This investigation focuses on the adaptive planning module, which constitutes a part of the pedagogical module. The pedagogical module is comprised by four sub-modules: exams evaluator, psycholexia, analyzer, and planner. The first two collect information about the student; while the third analyzes this information and instructs the planner on the actions to be taken. The architecture of the pedagogical module is shown in Figure 5.

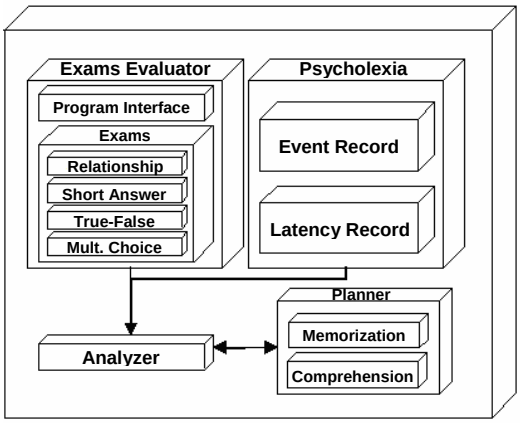


Figure 5: Pedagogical module architecture.

The exams evaluator sub-module is responsible for evaluating the student's learning degree of the tutoring system material, using five different types of assessment mechanisms: short answer exam, true-false answer exam, relationship exam, multiple choice exam, and external application exam.

The psycholexia sub-module is in charge of measuring the memorization and comprehension abilities and other elements of the students. The name of the module follows from one of the psychology areas, which studies the intellectual abilities of individuals concerning their quality. One of this discipline techniques (observation) was chosen in order to assess the students' cognitive abilities without intimidating them. Every observation is reported using records; one of which is *events*, used by the system for measuring the memorization and comprehension abilities. *Latency* is another type of record that the system uses for determining if the time devoted to study the material is a relevant factor in case the student shows a deficiency in his cognitive abilities, and other factors that will be described later on.

Finally, the planner module carries out the needed adaptations of the course contents, as well as reminding the student about needed review of some materials in order to make the learning process easier: all of this according to the instructions of the analyzer module. The planner sub-module has two mechanisms for adaptive planning of the contents: *memorization planning* and *comprehension planning*. Memorization planning in turn is constituted by two phases that remind and show the student the contents that he needs to review before studying a specific topic. Comprehension planning is more complex that the first one, and through this planning the system determines the number of slides that it will show to the student when studying each topic.

4.5 Assessment of the Cognitive Abilities

Since one of the important aspects of the tutoring system is to avoid intimidating the student when assessing his memorization and comprehension abilities, an extensive search of psychological techniques and methods was conducted to this end. The search revealed that there exists a technique in psycholexia called *observation*, which permits measuring the individuals' intellectual abilities without being aware that they are being evaluated. Observation is the most frequently used psychological technique and its fundamental value consists of being able to study individuals in a natural way; i.e., the observer gets in direct contact with the individual's behavior. Every observation uses different tools for recording the specific activities investigated. The instruments used by the tutoring system are *event recording* and *latency recording*.

Event recording consists of counting individual's behaviors as they occur. Recording the number of times that a behavior under investigation occurs is an observation recording technique frequently used. In our case the tutoring system uses such recording for counting the following values: number of memorization related questions and correct answers, number of comprehension related questions and correct answers, and ratio of slides studied.

Latency recording consists of measuring the time elapsed from the beginning of a stimulus (a command) to the beginning of a behavior. Latency recording is used when the main interest is in the amount of time between an opportunity to trigger a behavior and the beginning of the behavior execution. In our case the tutoring system uses this recording for obtaining the following values: rate of study of memorization related slides and rate of study of comprehension related slides.The values for the number of memorization related questions and correct answers, and the number of comprehension related questions and correct answers are obtained from the student's answers to the exams questions; while the rate of studied slides is obtained counting the number of slides examined in each student's session.

Finally, the rates of study of memorization and comprehension related slides are obtained calculating the time elapsed from the time the system displays a slide until the student requests an action, such as viewing the next or the previous slide. These variables are used by the tutoring system for determining if the student has been devoting enough time for studying the slides materials; and if the system detects cognitive abilities deficiencies, the system will recommend the student to devote more time for studying the slides.

4.6 Process of Adaptive Teaching Planning

Once the student's memorization and comprehension abilities have been assessed, the system has the required information for carrying out the adaptive planning of the material to be taught. As mentioned in Section 4.4, the planning is divided into two parts: memorization planning and comprehension planning.

The memorization planning permits the tutoring system to recommend the student reviewing a topic that the system considers has been forgotten and is necessary for the next topic. This planning consists of two phases. The first phase is carried out when the student accesses the course chapters and triggers the following sequence:

1. Get the student's memorization value (MemStu):

$$\text{MemStu} = \frac{\text{NumMemRite} \times 100}{\text{NumMemQstn}}$$

2. Get the average memorization value for all the students enrolled in the course (AvgMem).
3. **If** (MemStu < AvgMem) **then**:
   3.1. Get the rate of analysis of memorization related slides for the student (RAMSStu).
   3.2. Get the average rate of analysis of memorization related slides for all the students enrolled in the course (AvgRAMS).
   3.3. **If** (RAMSStu < (AvgRAMS) **then**:
      3.3.1. Display the message "SUGGESTION: study carefully the course material for improving learning."
   3.4. **End if**.
   3.5. Get the list of all the topics examined by the student such that number of reviews < recommended number of reviews (TopExaStud).
   3.6. Filter TopExaStud eliminating the topics that the student has reviewed in the preceding six days.
   3.7. **If** (TopExaStud.length > 1) **then**:
      3.7.1. Display the topics in the TopExaStud list.
   3.8. **End if**.
4. **End if**.

An illustration of the process is shown in Figure 6.

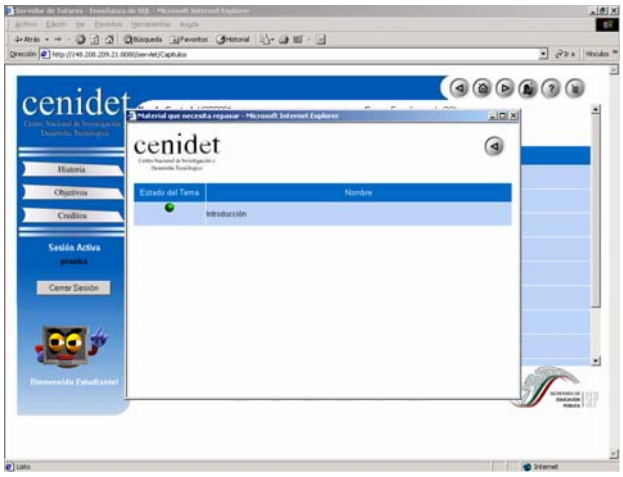


Figure 6: Illustration of the first phase of memorization planning.

The second phase of the memorization planning is carried out when the student starts to examine a topic. In this case the planner performs the following sequence:

1. Get the student's memorization value (MemStu):

$$\text{MemStu} = \frac{\text{NumMemRite} \times 100}{\text{NumMemQstn}}$$

2. Get the average memorization value of all the students enrolled in the course (AvgMem).
3. **If** (MemStu < AvgMem) **then**:

3.1. Get the rate of analysis of memorization related slides for the student (RAMSStu).
3.2. Get the rate of analysis of memorization related slides for all the students (AvgRAMS).
3.3. **If** (RAMSStu < AvgRAMS) **then**:
  3.3.1. Display the message "SUGGESTION: study carefully the course material for improving learning."
3.4. **End if**.
3.5. **If** there exists a prerequisite topic, which has memorization related slides and has been studied more that seven days ago, **then**:
  3.5.1. Get the slides of the prerequisite topic (PreTopSlid) and display them.
3.6. **End if**.
4. **End if**.

An illustration of this process is shown in Figure 7.

Figure 7: Illustration of the second phase of memorization planning.

Comprehension planning, unlike memorization planning, is applied to all students with more or less extent, except when the value of the student's comprehension ability equals the highest value (100 points) or when the student has exceeded the average number of slides that he examines in each session. Comprehension planning consists of adjusting the number of comprehension related slides (specifically explanation slides) that the system will present to the student; to this end the planner uses the following formula:

Where:

$$ES = \frac{\left((Com_{Max} - Com_{Stu}) \times \left(\left(\frac{(DD_{Topic} \times (\frac{NS}{2}))}{DD_{Max}} \right) + (\frac{NS}{2}) \right) \right)}{(Com_{Max} - Com_{Min})}$$

ES = Number of explanation slides to be displayed.
Com_{Max} = Highest possible comprehension value.
Com_{Stu} = Value of the student's comprehension.
Com_{Min} = Lowest possible comprehension value.
DD_{Topic} = Difficulty degree of the topic.
DD_{Max} = Highest possible difficulty degree of any topic (5 points).
NS = Number of explanation slides available for the topic.

This formula consists of two parts or filters that determine the number of explanation slides to be displayed to the student: the first filter (innermost part of the numerator) calculates the advisable number of slides according to the difficulty degree of the topic; while the second filter (outermost part) calculates the final number of slides to be displayed to the student according to the value of his comprehension ability and the value yielded by the first filter.

The comprehension planner carries out the following sequence:

1. Get the number of slides examined by the student (SlidExaStu).
2. Get the average number of slides that the student examines in each session (AvgSlid).
3. **If** (SlidExaStu < AvgSlid) **then**:
   3.1. Get the student's comprehension value (ComStu)

$$\text{ComStu} = \frac{\text{NumComRite} \times 100}{\text{NumComQstn}}$$

   3.2. Get the difficulty degree of the topic the student is about to examine (DD).
   3.3. Get the number of existing explanation slides for the topic the student is about to examine (NS).
   3.4. Calculate the advisable number of explanation slides to display to the student:

$$\text{ES} = \frac{\left((100 - \text{ComStu}) \times \left(\left(\frac{\text{DD} \times (\text{NS}/2)}{5}\right) + (\text{NS}/2)\right)\right)}{(100 - 0)}$$

   3.5. Display compulsory slides and DA explanation slides.
4. **End if.**

The following example explains this process. Consider a topic whose difficulty degree equals 2 and has two compulsory and seven explanation slides. If a student had a comprehension value equal to 100, the planner would not display explanation slides, as shown in Figure 8.

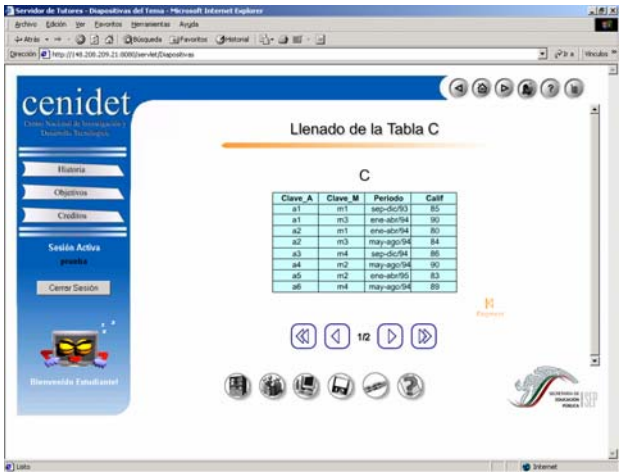


Figure 8. Illustration of planning for inactive comprehension (two slides will be displayed).

If a student had a comprehension value equal to 50, the planner would yield the following value:

$$\text{DA} = \frac{\left((100 - 50) \times \left(\left(\frac{2 \times (7/2)}{5}\right) + (7/2)\right)\right)}{(100 - 0)} = 3$$

(Note: all division results were rounded up to the closest integer.)

With this result the comprehension planner displays five slides (the two compulsory ones and three additional explanation slides), which the systems deems necessary according to the comprehension ability of the student and the difficulty degree of the topic that the student is about to examine. The result of the comprehension planning for this case is shown in Figure 9.

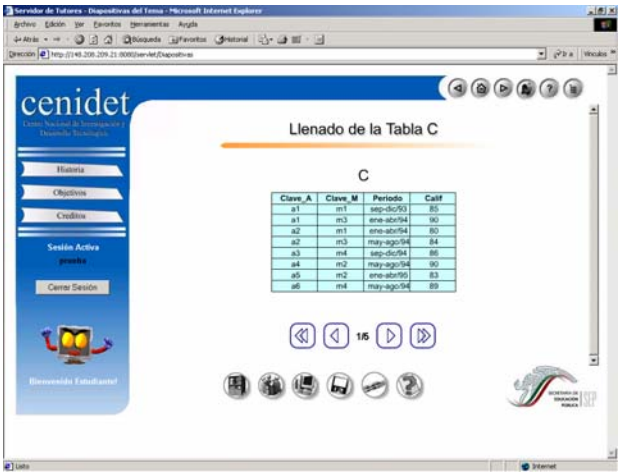


Figure 9. Illustration of planning for active comprehension (5 slides will be displayed).

5 EXPERIMENTAL TESTS

Figure 10. Interfaces of the tutoring systems.

In order to evaluate the benefit attained by incorporating into a tutoring system an adaptive planning that considers cognitive abilities, we used the PEMGU (Pedagogical Evaluation Methods Guidelines for Multimedia Applications). This test was developed by the European Union so that education institutions had a tool for evaluating the quality of education software [10].

The PEMGU test evaluates three main aspects:

a) **Technical and instructional adaptation**. This aspect evaluates the interface design, simplicity of information access and control, simplicity of usage, and documentation.
b) **Didactics or curricula adaptation**. This aspect evaluates curricula objectives handling, learning activities and materials, evaluation mechanisms for students, motivation factors, and application areas of the system.
c) **Global evaluation**. This aspect considers general features of the system when used by students.

Figure 10 shows the interfaces of the tutoring systems evaluated in the PEMGU test: on the left, the tutoring system without planning module [4]; and on the right, the tutoring system with the module for adaptive teaching planning.

The results of the tests are shown in Table 1.

Table 1. Results of the PEMGU test

No.	Element	Evaluation		
		System with planner	System without planner	Maximal value
Technical and instructional adaptation				
1	Window design	40	29	40
2	Information access & control	23	19	50
3	Use	15	10	15
4	Documentation	10	6	20
Didactics or curricula adaptation				
5	Learning objectives	9	0	60
6	Learning contents	72	65	75
7	**Learning activities**	**43**	**28**	**60**
8	**Evaluation**	**40**	**24**	**55**
9	Motivation	33	20	35
10	Program ease	19	14	30
Global evaluation				
11	Global evaluation	39	25	40
	TOTAL	**343**	**240**	**480**

As can be observed, the results show a slight improvement in the technical and instructional adaptation due to interface changes in order to make it friendlier. Regarding didactics or curricular adaptation, the test reveals significant improvements concerning the learning activities, which shows the success of the module for adaptive learning planning. This is so because adapting the course contents to the student's memorization and comprehension abilities permits to improve the learning process. Additionally, the evaluation mechanisms showed improvement, since the new version of the system includes more and better tools for evaluating knowledge acquisition by students.

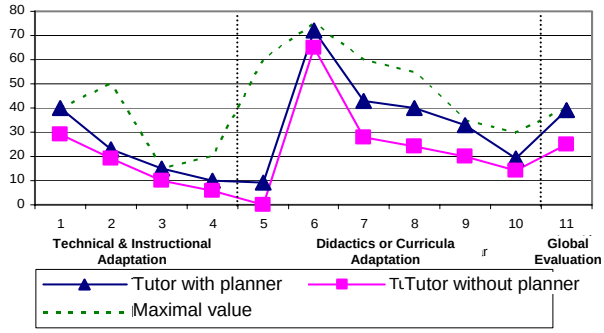


Figure 11. Graphical results of the PEMGU test.

Figure 11 shows the results of the PEMGU test graphically in order to highlight the most different aspects of both tutoring systems. The figure also includes the maximal values for each aspect of the PEMGU test.

6 CONCLUSIONS

Considering the cognitive abilities in the implementation of tutoring systems is not new, and some examples are the works by Julita Vassileva, Peter Brusilovsky and John Anderson [1, 11]. However, this project, unlike some of the aforementioned, includes the evaluation and involvement of students' cognitive abilities. Additionally, the adaptive planning module can support multiple courses and its operation is completely independent of the topic to be taught, which facilitates the authoring of quality tutoring systems.

Another important improvement consists of, not only considering the memorization and comprehension abilities in the planning of the material to be taught, but determining a possible cause for deficiency in one of the cognitive abilities such as the time devoted to study; thus allowing to improve the student's learning degree through suggestions.

7 REFERENCES

[1] Anderson J.R, "Cognitive Tutor: An Effective Way to Teach Math", www.significantdifference.org/Anderson.html, 2003.
[2] PACT Center, "Pittsburgh Advanced Cognitive Tutor Center", Carnegie Mellon University, www-2.cs.cmu.edu/~pact/.
[3] Ingram Ramírez J.W., *Herramienta para el Desarrollo de Sistemas Tutores*, MS thesis, Computer Sci. Dept. CENIDET, Cuernavaca, Mexico, 1999.
[4] Martínez Castro J.M., *Desarrollo de Módulos para Presentación Multimodal de Conocimientos y Acceso Vía Internet para un Sistema Tutor*, MS thesis, Computer Sci. Dept., CENIDET, Cuerna-vaca, Mexico, 2002.
[5] Howe D., "Interactive Courseware", wombat.doc.ic.ac.uk/foldoc/foldoc.cgi ?Interactive+CourseWare, 1995.
[6] Wenger E., *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann Publishers, 1987.
[7] Hartley J.R. and Sleeman D.H., "Towards Intelligent Teaching Systems", *International Journal Man-Machine Studies*, vol. 5, pp. 215-236, 1973.
[8] OMH, "Lidiando con Disfunciones Cognoscitivas Asociadas a Impedimentos Psiquiátricos", www.omh.state.ny.us/omhweb/spansite/cogdys_ manual/ CogDysHndbk.htm, 2002.
[9] Distance Learning, "Bloom´s Taxonomy", www. dlrn.org/library/dl/guide4.html, 2000.
[10] Sobrino A., Reparaz C., Santiago R., and Mir J.I., "Evaluación de Software Educativo: Propuesta de una Escala de Valoración OnLine", tecnologiaedu.us.es/edutec/paginas/ p2f2.htm, 2001.
[11] Brusilovsky P. and Vassileva J., "Course Sequencing Techniques for Large Scale Web-based Education", julita.usask.ca/homepage/Texte/BrusilovskyVassileva-print.pdf, 2003.

Objetos de Aprendizagem no Ensino de Lógica de Programação

Alexandre Negrão de Jesus, Daniel Lourenço Lopes,

Fernanda Rezende Perin, Juliana Martin Cantão, Edson Pinheiro Pimentel.

Universidade IMES
Av. Goiás, 3400, São Caetano do Sul, SP, Brasil, 09550-051

alexandrenegrao@micropower.com.br, d_llopes@yahoo.com.br,
fe_perin@hotmail.com, juliana_cantao@hotmail.com,
edson.pimentel@imes.edu.br

Abstract

In education environments to promote and to stimulate the use of educational resources make possible the improvement of the quality of education and the learning. The objective of this research is to present one brief introduction on the problems that the current education is passing and how the learning objects can assist the professors and students, presenting the classifications, characteristics and standards of objects of learning as well as advantages of its use and used object examples of learning already as a tool of pedagogical aid. Moreover, this paper presents a proposal to use learning objects in Programming Logic teaching and learning.

Resumo

Em ambientes de educação, promover e estimular o uso de recursos educacionais possibilita a melhoria da qualidade do ensino e do aprendizado. O objetivo desta pesquisa é apresentar uma breve introdução sobre os problemas que a educação atual está passando e como os objetos de aprendizagem podem auxiliar os professores e alunos, apresentando as classificações, características e padrões dos objetos de aprendizagem bem como vantagens de sua utilização e exemplos de objetos de aprendizagem já utilizados como uma ferramenta de auxílio pedagógico. Além disso, este artigo apresenta uma proposta de uso de objetos de aprendizagem no ensino de lógica de programação.

1. INTRODUÇÃO

A educação nos dias atuais está passando por um processo de renovação de espaços, de re-significação de conteúdos e de valores, tendo como ponto de partida todas as mudanças ocorridas na sociedade. A escola, como instituição integrante e atuante dessa sociedade e desencadeadora do saber sistematizado, não pode ficar fora ou a margem deste dinamismo (Maria, 2006).

De acordo com a Rived (2006) as possibilidades que se apresentam no conhecimento e no aprendizado, com o uso das Tecnologias de Informação e Comunicação (TIC), estão provocando alterações nas relações entre as competências intelectuais e o conhecimento, oportunizando mudanças de paradigma em situações de aprendizagem. Isso afeta diretamente as relações de ensino e aprendizagem e as modalidades de educação. Neste contexto, os objetos de aprendizagem se apresentam com possibilidades de potencializar o processo ensino e aprendizagem significativa dos conteúdos. Desponta na educação presencial e a distância como uma tecnologia recente e que pode beneficiar professores e alunos, nos ambientes de aprendizagem.

As problemáticas e dificuldades pertinentes ao aprendizado de programação, principalmente nas disciplinas introdutórias dos cursos de Computação e Informática, constituem um desafio para os docentes. As dificuldades encontradas podem ser diagnosticadas não somente pelo alto grau de repetência, mas também pelos problemas demonstrados por alunos em disciplinas avançadas que exigem o pré-requisito de programação [Rocha, 1991].

Este artigo tem por objetivo apresentar a arquitetura de uma ferramenta para o ensino de lógica de programação com uso de Objetos de Aprendizagem Um objeto de aprendizagem permite ao aluno modificar seu conhecimento, facilitando a representação e manipulação do objeto de estudo de forma individualizada, permitindo a atuação em grupos na resolução dos problemas e a perspectiva gerencial para o acompanhamento sistemático dos projetos e comunicação professor - aluno (Koslosky, 1999).

O artigo está organizado como segue. A seção 2 apresenta o conceito e mostra exemplos de Objetos de Aprendizagem; a seção 3 apresenta a proposta de uma ferramenta para o ensino de Lógica de Programação com o uso de objetos de aprendizagem. Na seção 4, são feitas algumas considerações acerca deste trabalho e os aprofundamentos necessários.

2. OBJETOS DE APRENDIZAGEM

De acordo com Colello (2006) atualmente, os educadores vêm se dando conta de que o problema de aprendizagem remete a outras dimensões dentro e fora da escola, propondo a instituição de ensino em nova configuração e dinâmica de trabalho.

A educação escolar, em todos os níveis, sempre esteve presa a lugares e tempos determinados: salas de aula, calendário escolar, grade curricular, modelos pedagógicos centrados no professor, que procura ajudar a que cada um consiga avançar no processo de aprender. Mas tem os limites do conteúdo programático, do tempo de aula, das normas legais, e por conta disto boa parte dos professores é previsível, não surpreende os alunos, repete fórmulas e sínteses (Moran, 2003).

E se os professores pudessem ensinar com mais eficiência e os alunos aprendessem brincando? Partindo de exemplos práticos para despertar a curiosidade dos estudantes, os objetos de aprendizagem têm se tornado uma importante ferramenta para aprimorar o ensino interativo de temas didáticos das mais diversas disciplinas.

O Objeto de Aprendizagem é definido como uma entidade, digital ou não-digital, que pode ser usada, re-usada ou referenciada durante o ensino com suporte tecnológico. Exemplos

de ensino com suporte tecnológico incluem sistemas de treinamento baseados no computador, ambientes de aprendizagem interativa, sistemas instrucionais auxiliados por computador, sistemas de ensino a distância e ambientes de aprendizagem colaborativa. Exemplos de Objetos de Aprendizagem incluem conteúdo multimídia, conteúdos instrucionais, objetivos de ensino, software instrucional e software em geral e pessoas, organizações ou eventos referenciados durante um ensino com suporte tecnológico (Wiley, 2000).

Objetos de Aprendizagem são elementos digitais fundamentados no conceito da orientação a objeto, segundo Wiley (2000). De acordo com Downes (2001), a idéia de orientação a objetos tende ao desenvolvimento de protótipos reais que, uma vez definidos, são copiados e usados por uma parte do software, quando necessário. Porém, através desta definição, entende-se que o conceito de orientação a objetos atua somente em ambientes digitais. Uma adaptação desta definição pode ser dita como sendo um protótipo de uma entidade. Uma vez definida, poderá ser copiada e usada em outros ambientes, digitais ou não.

Um dos ambientes digitais para disposição de Objetos de Aprendizagem são os Sistemas de Gerenciamento de Ensino (Learning Management System – LMS), que são sistemas de treinamento auxiliados por tecnologias, são sistemas interativos que abrigam cursos e são destinados ao apoio ensino-aprendizagem.

De acordo com Wiley (2000), todos os objetos de aprendizagem possuem certas características críticas e a forma como cada um deles as exibe faz um tipo de objeto de aprendizagem ser diferente de outro. Seu objetivo, com esta afirmação, é diferenciar os possíveis tipos de objetos de aprendizagem disponíveis para uso em *design* instrucional.

A figura 1 apresenta um exemplo de objeto de aprendizagem extraído de Fujii (2007). Por exemplo, a partir de um objeto atômico (quadro da Mona Lisa) pode se organizar um módulo ou curso sobre História da Arte.

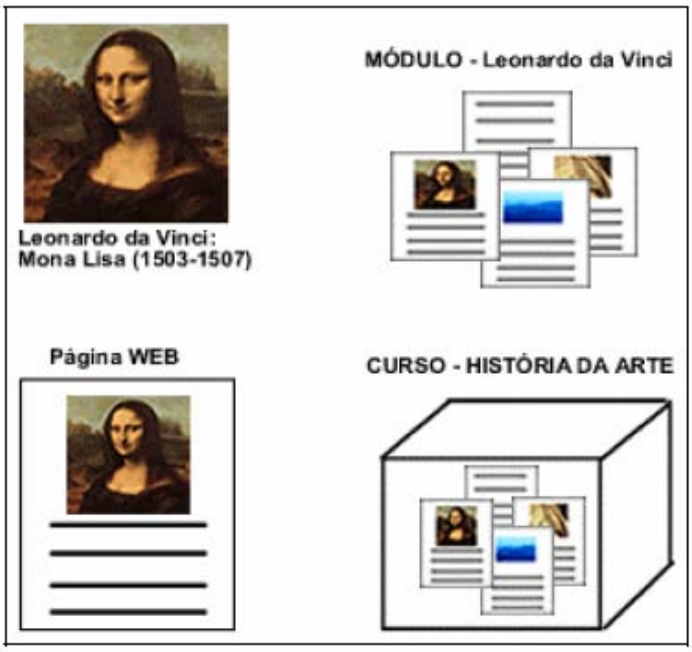


Figura 1. Exemplo de objetos de aprendizagem extraído de Fujii (2007).

A seguir a descrição de dois objetos de aprendizagem concretos:

- O objeto de aprendizagem "Nós no mundo" (figura 2) foi produzido por uma equipe de pesquisadores das áreas de educação, informática e informática na educação dos programas de pós-graduação em educação e informática na educação da Universidade Federal do Rio Grande do Sul. A equipe foi desafiada a desenvolver um objeto de aprendizagem que pudesse ser utilizado com alunos do 1º ciclo do ensino fundamental. (Konrath, 2006).]

Figura 2. Objeto de Aprendizagem "Nós no Mundo'

- O objeto da figura 3 é destinado ao uso por alunos de ensino médio. Os fenômenos são modelados através da animação interativa que simula um acontecimento da vida real, através de equações aceitas pela comunidade científica, no entanto, essas matemáticas não aparecem explicitamente. Considerando uma situação do cotidiano, o aluno é levado a refletir sobre as relações entre forças que atuam num corpo, e o possível movimento resultante. (Tavares, 2005).

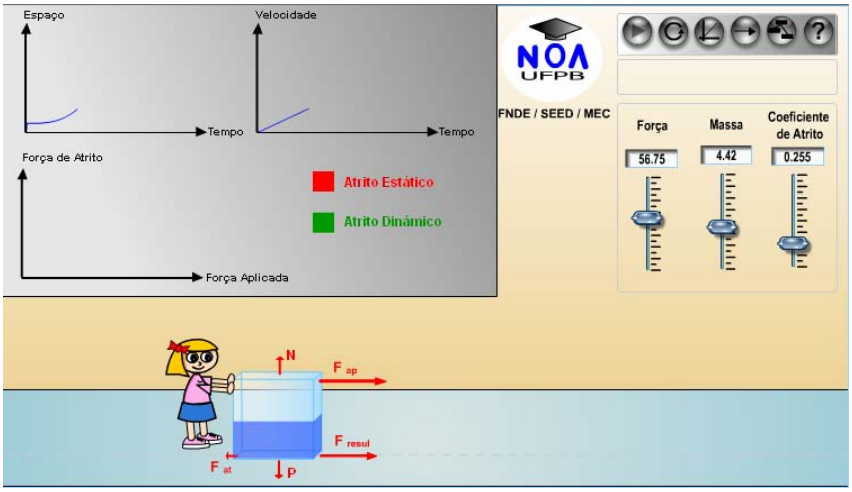


Figura 3. Objeto de Aprendizagem "Forças - Plano Horizontal"

3. ENSINO DE LÓGICA DE PROGRAMAÇÃO

Devido ao grande índice de dificuldade que os alunos dos cursos de computação encontram na disciplina de lógica de programação (ou nome similar), pretende-se oferecer uma ferramenta que utilizando objetos de aprendizagem possa auxiliar os alunos a aprender lógica de programação. Sem valorizar qualquer linguagem de programação, a ferramenta irá ensinar aos alunos, lógica de programação, utilizando português estruturado. Dessa forma os alunos aprenderão a lógica de uma maneira geral, sem se viciar em alguma linguagem específica.

A ferramenta será baseada nas metodologias existentes que apresentam os melhores resultados na absorção do conteúdo, psicologia cognitiva e nas técnicas utilizadas por professores da disciplina de lógica de programação (ou com nome similar) para auxiliar os alunos a adquirir conhecimento de uma maneira simples e de fácil absorção. Os assuntos abordados na ferramenta serão: Algoritmos, variáveis, operadores, estrutura de seleção e estrutura de repetição.

A ferramenta irá utilizar a estrutura apresentada na figura 4 para organização dos objetos de aprendizagem:

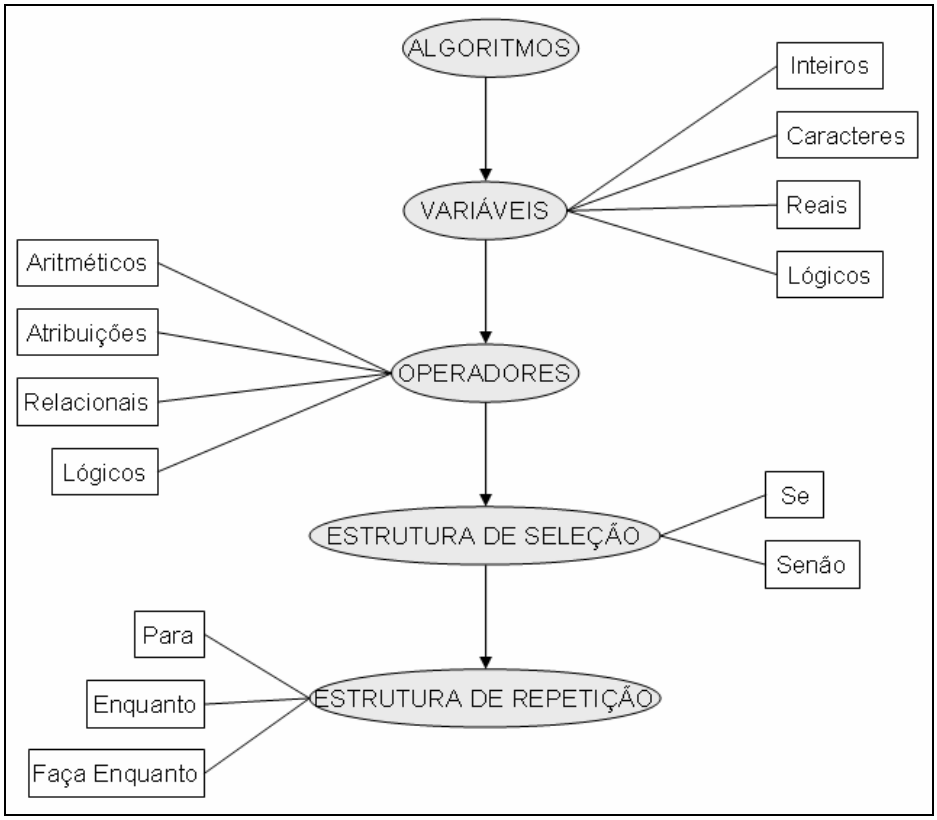


Figura 4. Estrutura do Curso e Organização dos Objetos de Aprendizagem

A figura 5 apresenta um exemplo de objeto de aprendizagem sobre o conceito de Fluxograma num curso de Lógica de Programação.

Figura 5. Exemplo do objeto de aprendizagem Fluxograma.

O ambiente para a criação e realização de cursos deverá permitir a inclusão de objetos interativos com animações, textos e imagens que motivem os alunos a utilizá-la e que sejam adaptados ao seu perfil. A figura 6 apresenta um esquema com a seqüência de uso do ambiente por parte do estudante.

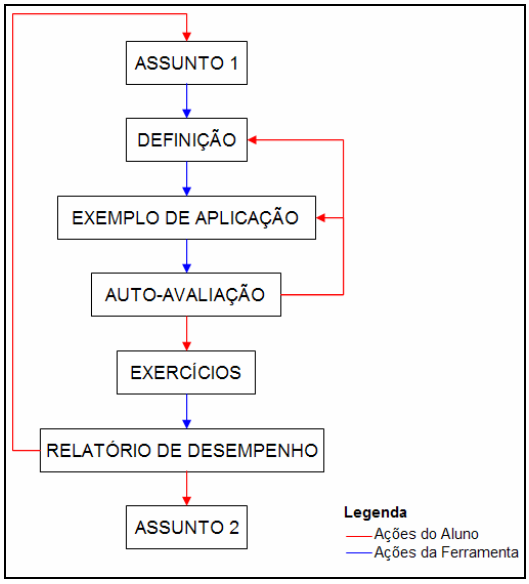


Figura 6. Esquema de Interação do aluno com a ferramenta

Para cada assunto, a ferramenta irá disponibilizar os seguintes itens:

- Definição: Explicação do assunto sobre: apresentação, principais conceitos e definições.
- Exemplo de Aplicação: Exemplo detalhado de onde o assunto poderá ser utilizado, onde o aluno poderá compreender de forma prática o assunto estudado.

- Auto-Avaliação: Permite ao aluno avaliar seu desempenho e caso for necessário, retornar nos itens de definição e/ou exemplificação do assunto para melhor entendimento.
- Exercícios: Após o estudo de um determinado assunto o aluno deverá realizar uma avaliação de múltipla escolha. Para poder estudar o próximo assunto o aluno deverá obter 70% de acerto na avaliação.
- Relatório de Desempenho: Permite ao aluno verificar seu desempenho nos exercícios. Para cada questão incorreta a ferramenta disponibilizará um relatório detalhado para cada erro, explicando a alternativa correta.

4. CONCLUSÕES

O domínio de conhecimento de "Lógica de Programação", em cursos de graduação em Computação e Informática, pela dificuldade de aprendizagem dos alunos, visível nos altos índices de reprovação, tem demonstrado ser um terreno fértil para pesquisas que busquem o aumento dos padrões de aprendizado através da identificação pontual das lacunas de aprendizagem, do monitoramento dos conhecimentos prévios e do acompanhamento contínuo da aprendizagem.

O uso de objetos de aprendizagem na educação é visto como uma técnica muito eficiente pois possibilita modularizar o curso até um nível atômico de forma a tratar cada conceito como indivisível. Em conjunto com os mapas conceituais torna-se possível criar uma abordagem para o uso de objetos adaptativos de aprendizagem apoiado na teoria de grafos.

Como trabalhos futuros pretende-se concluir a implementação da ferramenta e realizar estudos de casos reais em disciplinas de programação

REFERÊNCIAS

BETTIO, Raphael Winckler, MARTINS, Alejandro (2002). Objetos de Aprendizado: Um novo modelo direcionado ao Ensino a Distância.Cortez, São Paul

COLELLO, Silvia M. Gasparian (2006). "A Formação de Professores na Perspectiva do Fracasso Escolar". Disponível em: http://www.fe.usp.br/psicologia/silvia_escolar.doc Acesso em: Junho/2007

DOWNES, Stephen (2001). "Learning Objects: Resources for Distance Education Worldwide". IN: Internacional Review of Research in Open and Distance Learning. Vol. 2, No. 1. Disponível em: http://www.irrodl.org/content/v2.1/downes.pdf Acesso em: Junho/2007

FUJII, Noemi P. N. (2007). *Uma proposta de Objetos de Aprendizagem Reutilizáveis Adaptativos para o Ensino de Estatística*. Dissertação de Mestrado. UnicSul, São Paulo.

KOLOSKY, Marco Antônio Neiva (1999). "Aprendizagem Baseada Em Casos: Um Ambiente Para Ensino de Lógica de Programação" Universidade Federal De Santa

Catarina. Disponível em: http://www.eps.ufsc.br/disserta99/koslosky Acesso em: Maio/2007

KONRATH, Mary Lúcia Pedroso (2006). "Nós no Mundo: Objeto de Aprendizagem - voltado para o 1º Ciclo do Ensino Fundamental". Disponível em: http://www.cinted.ufrgs.br/renote/jul2006/artigosrenote/a2_20139.pdf Acesso em: Junho/2007

LONGMIRE, W. A . (2001). Primer On Learning Objects. American Society for Training & Development. Virginia. USA.

MARIA, Ana e Gabriela (2006). "A Tecnologia Renovando o Processo Educativo". Núcleo de tecnologia Educacional de Barreiras, BA Disponível em: http://edutec.net/Textos/Alia/PROINFO/prf_txtie16.htm Acesso em: Junho/2007

MORAN, José Manuel (2003). "Mudar a forma de ensinar e de aprender com tecnologias: Transformar as aulas em pesquisa e comunicação presencial-virtual". Disponível em: http://www.irrodl.org/content/v2.1/downes.pdf Acesso em: Junho/2007

RIVED (2006). "Planejamento e Desenvolvimento do Objeto de Aprendizagem". Disponível em: http://www.rived.mec.gov.br/artigos Acesso em: Junho/2007

ROCHA, Helena. V. (1991). Representações Computacionais Auxiliares ao Entendimento de Conceitos de Programação. Unicamp.

TAVARES, Romero (2005). "Objeto de Aprendizagem: Forças – Plano". Disponível em: http://www.fisica.ufpb.br/~romero/objetosaprendizagem/3aForcasPlanoHorizontal/ Acesso em: Junho/2007

WILEY, D. A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. The Instructional Use of Learning Objects: Online Version. In D. A. Wiley (Ed.). Disponível em: http://reusability.org/read/chapters/wiley. doc. Acesso em: Maio/2007

OLPC en Argentina: Evaluación de usabilidad frente a alumnos de nivel primario

F. Javier Diaz

Fac.de Informática, Universidad Nacional de La Plata.
La Plata, 1900, ARGENTINA
jdiaz@info.unlp.edu.ar

Claudia M. Tzancoff Banchoff

Fac.de Informática, Universidad Nacional de La Plata.
La Plata, 1900, ARGENTINA
cbanchof@info.unlp.edu.ar

Viviana Harari

Fac.de Informática, Universidad Nacional de La Plata.
La Plata, 1900, ARGENTINA
vharari@info.unlp.edu.ar

e

Ivana Harari

Fac.de Informática, Universidad Nacional de La Plata.
La Plata, 1900, ARGENTINA
iharari@info.unlp.edu.ar

Abstract

For the last several years, the Computer Science Faculty of the UNLP has been working in the dissemination and usage of open source software, emphasizing its usage in the educational sector.

During 2006, active work using Lihuen GNU/Linux has been done in several schools, and the outcome of these experiencies were presented in several meetings and conferences. Last February, two prototypes of OLPC were obtained, generating a new line of work which involves two main areas:

- on one hand, the evaluation of alternatives for the operating system proposed in the original version, Sugar.
- on the other hand, a series of evaluations which includes everything from a formal evaluation of the OLPC user interface, to a set of usability tests with children and teachers.

The objective of this article is to specifically analyze the first experience with children from one school, located at the City of La Plata, during their first contact with OLCP. Therefore, there will be presented the characteristics of the usability test that was performed, and its results.

Keywords: Software freedom, OLPC, Sugar, Lihuen GNU/Linux, Usability Testing

Resumen

Desde hace varios años la Facultad de Informática de la UNLP ha estado trabajando en la difusión y uso de software libre, haciendo principal hincapié en el sector educativo.
Durante el año 2006, se trabajó activamente en varias escuelas usando Lihuen GNU/Linux y los resultados de estas experiencias fueron expuestos en diferentes jornadas y encuentros.
En febrero del corriente año, se consiguieron dos prototipos de OLPC y se abrió una nueva línea de trabajo que involucra dos grandes ramas:

- por un lado, la evaluación de alternativas al sistema operativo Sugar, propuesto en la versión original.
- por el otro, una serie de evaluaciones que van desde una evaluación formal de la interfaz de usuario de las OLPC hasta una serie de testeos de usabilidad con niños y docentes.

El objetivo de este artículo, es analizar específicamente, la primera experiencia con niños de una escuela de la ciudad de La Plata, en su primer contacto con las OLPC. A través del mismo, se va a exponer las características del testeo de usabilidad realizado y los resultados obtenidos.

Palabras claves: Software libre, OLPC, Sugar, Lihuen GNU/Linux, Testeo de Usabilidad

1 Introducción

El proyecto OLPC nace en el año 2005, en el Media Lab de MIT [1] (Laboratorio de Medios del Instituto Tecnológico de Massachussets, EEUU), donde se planteó desarrollar una computadora portátil que se pudiese adquirir a muy bajo costo y que permitiese cambiar la forma en la que se educa. Una de sus características más importantes, es que las computadoras deben ejecutar únicamente software libre. Es así que nace la organización sin fines de lucro denominada "One Laptop per Child (OLPC - Una Laptop por Chico)" [2] que es independiente del MIT.

El proyecto OLPC, originalmente, contempla la distribución de los equipos por fuera del circuito mercantil. Los gobiernos interesados en participar, podrán comprar las computadoras y luego distribuirlas en forma gratuita a los alumnos de las escuelas.

A través de Educ.ar [3] se han podido conseguir los primeros prototipos de las OLPC y esto dio lugar a una línea de trabajo dentro del proyecto de Software Libre en las Escuelas que se desarrolla en la Facultad de Informática.

Este artículo describirá la primera experiencia llevada a cabo con niños frente a las OLPC. Esta tarea se realizó en la Escuela Anexa de la Universidad de La Plata [4] con niños de 5to. y 6to. grado de la Educación Primaria Básica (EPB). Se detallarán las características del testeo de usabilidad que se llevó a cabo, teniendo en cuenta sus objetivos principales, el perfil de los actores, las actividades programadas en el trascurso de la evaluación, el seguimiento y registro de los resultados observados. Finalmente, se incluirá un análisis de los resultados del testeo y se podrá observar que los mismos fueron altamente positivos.

2 Características del proyecto

La Argentina es uno de los primeros países en adherir al proyecto OLPC, y hasta el momento, el único de habla hispana, aunque hay otros paises evaluando su adopción.

Se ha destinado fondos para su evaluación por parte de especialistas en tecnología y educadores. La última etapa de este proceso contempla una serie de pruebas piloto, la última de las cuales se realizará en las escuelas con los mismos alumnos.

El Ministerio de Educación de la Nación, a través de Educ.ar ha organizado y coordinado diferentes actividades con distintos equipos de trabajo de manera tal de poder evaluar adecuadamente el proyecto y su implementación en nuestro país.

Es a través de Educ.ar que se han podido conseguir los primeros prototipos para comenzar a trabajar.

El proyecto OLPC en la UNLP está siendo llevado a cabo dentro del Laboratorio de Investigación de Nuevas Tecnologías en Informática, LINTI [5]. El LINTI funciona dentro de la Facultad de Informática de la Universidad Nacional de La Plata y es uno de los tres laboratorios de investigación con que cuenta esta Unidad Académica.

Después de varias gestiones, en julio de 2006 se pudo acceder a los primeros prototipos de OLPC. Se trató de 2 plaquetas (ver Figura 1) cuyas características técnicas se muestran en la Tabla 1.

Características	Detalle
Procesador	AMD Geode de 366 Mhz
Memoria RAM	128Mb
Memoria Flash	512Mb
Display	--
Puertos USB	3
Conectividad	Wireless 802.11b/g compatible
Sistema Operativo	--

Tabla 1: Características del primer prototipo de OLPC.

Figura 1: Imagen del primer prototipo de OLPC

A partir de este logro, dentro del Programa de Investigación que se mantiene en el LINTI sobre "Software Libre"[6], se abrió una nueva línea de investigación para realizar una serie de pruebas e investigar su potencialidad de uso en la realidad educativa de nuestro país, considerando todas las alternativas posibles de implementación.

Como se mencionó anteriormente, una de las características más destacadas del proyecto OLPC es que las computadoras se distribuirán con software libre. En particular, el grupo de desarrollo de este proyecto propone la utilización de un sistema operativo Fedora Core con un escritorio especialmente escrito para estas computadoras denominado Sugar [7].

En febrero de este año, se consiguieron dos prototipos completos (ver Figura 2) con la apariencia de la portatil que se distribuiría entre los alumnos. Las características técnicas de este modelo pueden verse en la Tabla 2.

Características	Detalle
Procesador	AMD Geode de 366 Mhz
Memoria RAM	240 M
Memoria Flash	512Mb
Display	pantalla LCD de 7.5 pulgadas
Conectividad	Wireless 802.11b/g compatible
Puertos USB	3
Sistema Operativo	Linux Fedora Core

Tabla 2: Características del segundo prototipo de OLPC

Figura 2: Imagen del segundo prototipo de OLPC

Con la adquisición de estos prototipos, se plantearon varias líneas de trabajo que abarcan desde el desarrollo de software, a la evaluación de los mismos en entornos reales. Estas líneas fueron descriptas y publicadas en [8].
Una primer línea de investigación se refiere al desarrollo de la distribución local basada en Lihuen GNU/Linux para las OLPC [9].

Otra línea de trabajo se refiere a la evaluación heurística de la interfaz de usuario propuesta. La evaluación heurística [10] es uno de los métodos principales de inspección y se lleva a cabo por expertos en HCI, Interacción Hombre-Computadora. El objetivo es evaluar la calidad de uso de un producto desde le punto de vista del diseño de la interfaz.

Por último, se formó un grupo de trabajo que se focaliza en el estudio de factibilidad áulica de las OLPC, teniendo en cuenta las actividades que realizan los docentes con sus alumnos y el rol de la PC en la clase.

3 Primer Evaluación de Usabilidad

Como parte del proyecto de la evaluación de las OLPC, se organizó un testeo de usabilidad donde se puso a prueba el producto frente a usuarios reales, en este caso los niños.

Se escogieron alumnos pertenecientes a la Escuela Anexa "Joaquin V. Gonzalez", escuela básica pública dependiente de la Universidad Nacional de La Plata. Esta escuela es una de las instituciones en dónde se viene trabajando con la difusión y el uso de aplicaciones de software libre en el aula.
La escuela cuenta con una sala de PC con 20 máquinas, de las cuales 16 son destinadas para el uso de los alumnos. En general trabajan de a dos por máquina.
El único sistema operativo utilizado hasta el año 2006, era Microsoft Windows, con aplicaciones basadas en dicho sistema. A partir del año 2007, se ha incorporado un sistema operativo de software libre: Lihuen GNU/Linux [11], por lo tanto todas las máquinas pueden arrancar con ambos sistemas.
Al gabinete de computación, asisten periódiamente alumnos a partir de los 5 años (Nivel Inicial, 3er sala), para realizar tareas que permitan afianzar los conceptos abordados por el docente en las diferentes disciplinas.

3 1 Perfil del alumnado

Los alumnos seleccionados, a los cuales se le realizó el test de usabilidad, pertenecen a los últimos grados del nivel 2 de EPB (Educación Primaria Básica) y sus edades oscila entre los 10 y 12 años.
Los niños ya tenían una formación informática básica dado que asisten periódicamente a la sala de computadoras de la escuela, para realizar tareas relacionadas con los temas abordados en el aula. Dichas tareas están realizadas con diferentes aplicaciones, tanto de ofimática, como de uso de Internet y educativas.

3 2 Organización del test

Se seleccionaron en total 20 alumnos, los cuales trabajaron en grupos de a dos, dado que se contaban sólo con dos prototipos de máquinas OLPC.
Se diagramaron actividades de formato similar a las que los alumnos acostumbran a desarrollar con los entornos de Microsoft Windows y Lihuen GNU/Linux. En total fueron tres actividades con diferentes niveles de complejidad y que debían ser desarrolladas en una sesión de aproximadamente 20 minutos.
Durante el desarrollo del test se encontraron presente dos personas encargadas del seguimiento, observación de la interacción niño-OLPC y registro de resultados.
Antes de comenzar con las actividades, se les explicó a los alumnos las características básicas de la interfaz Sugar, describiendo brevemente la funcionalidad de cada una de las aplicaciones con las que cuenta la OLPC y se les mostró cómo hacer para ejecutar y cerrar una aplicación. Se les explicó brevemente el uso del editor de textos, una aplicación denominada Write, basada en el editor de textos Abiword [12], dándoles un pantallazo general sobre su uso, información básica necesaria para iniciar y desarrollar el test.

3 3 Actividades planteadas

A continuación se detallan las actividades realizadas por los niños.

Actividad 1:

- Abrir el editor de textos y escribir unas líneas dando su opinión respecto a las OLPC.
- Cambiar el tipo y tamaño de letra utilizada, respetando mayúsculas y acentos.
- Guardar el documento, colocándole un nombre deseado.
- Finalizar la tarea cerrando la aplicación

Actividad 2:

- Abrir nuevamente el editor de textos e insertar una tabla con tres columnas y dos filas. En la primer fila colocar las palabras: Grave, Aguda y Esdrújula
- Escribir en la columna correspondiente las palabras dictadas según su tipo de acentuación
- Agregar filas en caso de ser necesario.
- Grabar el archivo.
- Cerrar la aplicación

Actividad 3:

- Abrir la aplicación “Cámara”.
- Tomarse una foto.
- Guardar la imagen.
- Cerrar la aplicación.
- Abrir el editor de textos e insertar como imagen la foto adquirida en la aplicación anterior.
- Colocar utilizando viñetas información sobre su persona: nombre, apellido, edad, colegio, etc.
- Guardar el documento.
- Cerrar la aplicación

3 4 Seguimiento

Los chicos fueron monitoreados por dos personas que registraron sus acciones y observaron en forma directa cómo los alumnos interactuaban con el sistema. Durante este seguimiento, se trató de asistir al alumno lo mínimo posible, para ver si podía realizar las actividades con el conocimiento informático que ya tenían. La Figura 3 muestra una foto tomada al momento de la evaluación.

Figura 3: Alumnos de la Escuela Anexa interactuando con las OLPC

Los alumnos interactuaron con las OLPC, sólo una vez. Está previsto repetir esta experiencia con la misma población y chequear si recuerdan cómo utilizarla, realizando tareas similares. De esta manera se analizaría si su performance mejora o no.

4 Resultados de la experiencia

Como resultado de esta evaluación, se pudo registrar dos tipos de información. Por un lado, se pudo obtener valores concretos sobre la interacción de los niños con las OLPC, se pudo observar el comportamiento de los mismos en la sesión de trabajo que tuvieron y cómo fue su performance frente al sistema. Por otro lado, se pudo obtener información subjetiva, o sea se logró en el mismo encuentro, y a través de una de las actividades que los niños debían realizar con la máquina (actividad 1), tener un feedback por parte de los mismos, en las cuales se rescataron opiniones, críticas, si les gustó o no, y cierto análisis comparativo con el ambiente operativo tradicional que utilizaban hasta el momento.

Respecto a la interacción niño-OLPC, el test de usabilidad arrojó los siguientes resultados, que se van a mostrar en tablas organizadas por tarea. Se tuvo en cuenta si lo realizó o no y el grado de dificultad.

Actividad 1: *En el editor de textos, escribir su impresión sobre la OLPC*			
Tareas de la actividad 1	**% de alumnos que realizaron la tarea**	**% de alumnos que realizaron la tarea con dificultad**	**% de alumnos que no realizaron la tarea**
Ingresar a la aplicación	90%	10%	--
Utilización de mayúsculas y acentos en el texto	40%	60%	--
Cambio de tipo y tamaño de letra	60%	40%	--
Guardar el documento	80%	20%	--
Cerrar la aplicación	95%	5%	--

Actividad 2: *En el editor de textos, insertar una tabla y colocar dentro de ella palabras dictadas*			
Tareas de la actividad 2	**% de alumnos que realizaron la tarea**	**% de alumnos que realizaron la tarea con dificultad**	**% de alumnos que no realizaron la tarea**
Ingresar a la aplicación	100%	--	--
Creación de la tabla	95%	5%	--
Inserción de filas	60%	40%	--
Guardar el documento	100%	--	--
Cerrar la aplicación	100%	--	--

Actividad 3: *Utilización de la aplicación cámara para sacarse una foto, uso del editor de textos para insertar la imagen y colocar utilizando viñetas características personales*			
Tareas de la actividad 3	**% de alumnos que realizaron la tarea**	**% de alumnos que realizaron la tarea con dificultad**	**% de alumnos que no realizaron la tarea**
Ingresar a la aplicación cámara	90%	10%	--
Sacar foto	95%	5%	--
Ingresar a la aplicación editor de textos	100%	--	--
Encontrar icono de Insertar imagen	70%	30%	--
Encontrar la foto dentro del directorio	20%	80%	--
Insertar viñetas	70%	30%	--
Guardar el documento	100%	--	--
Cerrar la aplicación	100%	--	--

A continuación, se va a mostrar una tabla con un gráfico estadístico, que resume cómo fueron desarrolladas las tareas. Se muestra el grado de realización de las tres tareas, con o sin dificultad. Vale aclarar que para obtener un porcentaje de realización por actividad, como las mismas están compuestas por varios pasos, se promediaron los porcentajes de realización de todos ellos.

Actividad	***Porcentaje de realización sin dificultad***	***Porcentaje de realización con dificultad***
Actividad 1	*73*	*27*
Actividad 2	91	9
Actividad 3	81	19

En la Figura 4, se muestra los valores de la tabla en forma gráfica.

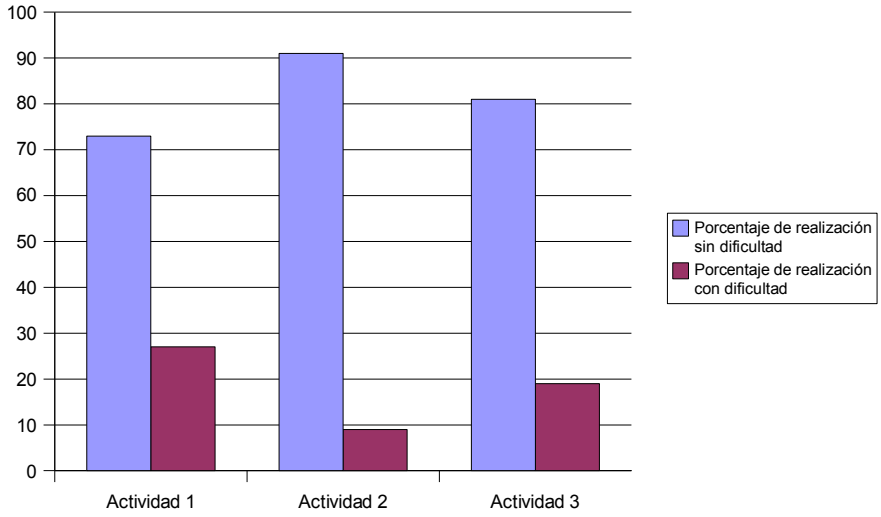


Figura 4: Porcentaje de realización de las actividades

Respecto a la evaluación de aspectos subjetivos, se tuvieron en cuenta cinco cuestiones principales: la aceptación general, la facilidad de uso, el uso del mouse y el teclado, y el grado de preferencia de Sugar respecto de los sistemas operativos conocidos por los niños.
Esta información se obtuvo de la Actividad 1, en donde se les pidió a los alumnos que escribieran su opinión de la OLPC.
La tabla resultante se detalla a continuación:

Aspectos subjetivos sobre la OLPC obtenidos de la primer actividad			
Aspecto	**% de alumnos que contestaron afirmativamente**	**% de alumnos que contestaron negativamente**	**Comentarios**
Aceptación general	100%	--	La aceptación fue unánime
Fácil de utilizar en general	100%	--	No se les presentaron inconvenientes
Fácil utilización del mouse	50%	50%	Fue lo que más le costó
Fácil utilización del teclado	100%	--	Aunque hubo asistencia en algunos casos, declararon que les parecía fácil su uso
Interfaz mejor que la que acostumbran a utilizar	100%	--	A todos les pareció mejor que la interfaz del sistema que usan.

Otros comentarios relevantes tomados en el transcurso de la evaluación fueron los siguientes:

- "*Me parece mejor que Windows, porque es más visual, ya que se ven bien qué aplicaciones están andando*"
- "*La interfaz me gusta más porque está mejor ordenada*"

- "*Me gustaría que estuviera todo en castellano*"
- "*Faltan teclas en el teclado, como la de Suprimir, Fin, etc*"
- "*La pantalla me parece muy chiquita*"

5 Conclusiones

Los resultados que arrojó la evaluación de indagación, fueron altamente satisfactorios. La información que se logró registrar en el test fue relacionada con aspectos de usabilidad pero también se extrajeron datos subjetivos, como el grado de satisfacción, si les gustó a los participantes y sus opiniones.
Respecto a la interacción que los alumnos tuvieron con las OLPC, los resultados reflejados en la sección anterior, muestran claramente la buena performance que tuvieron los chicos frente a un ambiente de trabajo que nunca antes habían visto. Todas las tareas fueron desarrolladas, con muy bajo nivel de dificultad.
Con respecto a los datos subjetivos extraídos por parte de los niños, hubo muestras de gran satisfacción, aceptación del producto, y mayor preferencia en relación con el ambiente de Windows. Las características de portabilidad, color y tamaño adecuados que presentan las OLPC, favorecieron el hecho que los alumnos tuvieron una aceptación casi inmediata con la máquina. Y esto trajo como consecuencia, una buena predisposición desde el comienzo, realizando las tareas con gusto y entusiasmo.
Está previsto continuar con esta serie de testeos, tomando otros perfiles de alumnos. De esta manera, se podrá contar con datos respecto de niños sin experiencia en uso de computadoras, o sólo con experiencia en uso de aplicaciones tradicionales, por ejemplo.
También se prevee realizar pruebas con los docentes a cargo de los niños. Los resultados de estos test, sus impresiones y sugerencias son sumamente importantes, dado que ellos serán los encargados de plantear y crear actividades con estos equipos ni bien sean distribuidos.

6 Referencias

[1] http://www.media.mit.edu

[2] http://laptop.org

[3] http://www.educ.ar

[4] http://www.graduada.unlp.edu.ar

[5] http://www.linti.unlp.edu.ar

[6] http://sl.info.unlp.edu.ar

[7] http://wiki.laptop.org/go/Sugar

[8] **OLPC en Argentina, análisis de realidades y potencialidades**. J. Díaz, I. Harari, V. Harari, P. Amadeo, C. Banchoff Tzancoff. http://www.ing.unp.edu.ar/wicc2007/trabajos/TIAE/155.pdf

[9] http://lihuen.info.unlp.edu.ar/tiki-index.php?page=ProyectoOLPC

[10] **Usability Inspection Methods**. Jacob Nielsen, Robert Mack.

[11] http://lihuen.info.unlp.edu.ar

[12] http://wiki.laptop.org/go/AbiWord#Write_.28Collaborative_Word_Processor.29

Producción de conocimientos en las Instituciones Universitaria. Aportes para analizar el impacto de la educación a distancia en los sistemas de gestión de conocimiento de la Universidad

EugeniaMárquez, Hugo Santos Rojas, María Elena Bain, Patricia Yañez
Universidad Nacional de la Patagonia Austral
{emarquez, hrojas,mebain}@unpa.edu.ar; pyaniez@uaco.unpa.edu.ar

Resumen

Si la tecnología fuera considerada en la Educación a Distancia como ciencia aplicada, esto equivaldría a reducirla a un conjunto de "reglas tecnológicas"; a su vez, las reglas tecnológicas serían consecuencias deducibles de las "leyes científicas" y, por lo tanto, el "desarrollo tecnológico" dependería de la "investigación científica". Alternativamente, desde un enfoque sistémico se considera que el factor fundamental del desarrollo tecnológico es la innovación social y cultural, la cual involucra no solamente a las tradicionales referencias al mercado, sino también a los aspectos organizativos, y al ámbito de los valores y de la cultura. Este enfoque nos habilita a poder pensar a la tecnología como un sistema de relaciones en el cual estamos involucrados.
En su carácter pedagógico debe contemplar la innovación que constituye en principio la creación o adaptación de nuevos conocimientos y su aplicación. El análisis de la propuesta del Sistema de Educación a Distancia de la UNPA nos permitirá visualizar las relaciones entre los tres ejes: el pedagógico, el organizacional y el tecnológico.
Se pretende en una primera instancia, analizar los componentes de los modelos pedagógicos, tecnológicos y organizativos más significativos que surgen al aplicar el modelo unificado (UML) como metodología de desarrollo para la definición e implementación del sistema educativo Unpabimodal. En segundo término se busca evaluar si la incorporación del sistema de educación a distancia genera nuevas necesidades de información o por el contrario pone al descubierto la falta de una cultura institucional que valora la información de base y sostiene la posibilidad de producción de conocimiento.

Palabras Clave: Modelo Educativos Virtuales, Modelos Pedagógicos, Modelos Organizacionales, Gestión de Información, metodología de desarrollo.

Introducción

En la Argentina, se espera de las instituciones universitarias públicas un fuerte compromiso en la búsqueda de respuestas que posibiliten el desarrollo social, cultural, científico y económico. Se espera que se comprometa en la creación, desarrollo, transmisión e interpretación crítica de los fenómenos científicos y culturales; la formación de profesionales con conocimientos, competencias y métodos; el desarrollo de acciones que contribuyan a la mejora de la calidad de vida de los habitantes; y la generación de oportunidades para garantizar la formación de los ciudadanos a lo largo de la vida.

Este conjunto de demandas pueden ser consideradas o analizadas desde las funciones tradicionales de la Universidad (docencia, investigación y extensión) pero el factor novedoso es sin duda, el tiempo de respuesta y la calidad esperada, por parte del medio, para atender a dichas demandas. Esta situación y otras producen además, ciertos movimientos internos en las instituciones de referencia, promoviendo cambios en diversos sentidos incluyendo la redefinición de su estructura organizativa y sus dinámicas de funcionamiento.

Ante esta situación las organizaciones han ido incorporando a las Tecnologías de la Información y la comunicación, lo que les ha ido permitiendo dar respuestas por un lado, pero también han ido generando ciertas capacidades en los demandantes, para alcanzar su autonomía, organización y velocidad.

Estas tecnologías que sostienen lo que se denomina sociedad de la información, permitieron imaginar e incluso reclamar la constitución de lo que hoy denominamos sociedad del conocimiento.

La Universidad Nacional de la Patagonia Austral fue producto del proceso de transformación de las instituciones de educación superior ya existentes en la provincia de Santa Cruz, originado en el año 1987 a partir de la autoevaluación y de la reformulación institucional generadas en el Instituto Universitario de Santa Cruz y en los Centros de Estudios Terciarios ubicados en cuatro localidades de la Provincia.

Con la intención de articular los esfuerzos en la oferta de Educación Superior de la región los gobernadores de las provincias de Santa Cruz y de la entonces recientemente creada Tierra del Fuego, firmaron en el año 1990 un Tratado Interprovincial mediante el cual se creó la Universidad Federal de la Patagonia Austral (UFPA).

En el año 1993 se inició el trámite de nacionalización de la UFPA y, mediante la ley 24.446 sancionada el 23 de diciembre de 1994 y promulgada el 11 de enero de 1995 se creó la actual Universidad Nacional de la Patagonia Austral sobre la base de la universidad provincial.

La Universidad, desde su creación, atravesó por dos procesos de Evaluación Institucional, el último de ellos en el año 2002. A partir de los diagnósticos surgidos de este último proceso y del debate en los órganos de gobierno, la UNPA preparó un Plan de Desarrollo Institucional plurianual.

Por otro lado, la Universidad constituyó en el año 2000 el Programa de Educación a Distancia y el Programa de Sistemas y tecnologías de la Información (PSTI).

De acuerdo con Salinas (2004b) lo importante no está en si es necesario o no nuevos modelos, sino cómo se tienen que combinar los componentes o elementos pedagógicos, organizativos y tecnológicos del entorno de aprendizaje. Para ello es necesaria la

experimentación y la validación de modelos pedagógicos que expliquen y permitan una adecuada explotación de los entornos virtuales de formación.
Es por tanto indispensable conocer qué dimensiones abarca y qué componentes forman estos procesos de E-A para poder centrar nuestra atención en aquellos elementos que nos interesan y plantear modelos que se adapten a la diversidad de situaciones surgidas con la aparición de las TIC. El proceso de experimentación y validación de nuevos prototipos nos permitirá conocer, no sólo el tipo de configuración tecnológica requerida, si no también sistemas de implantación, metodologías a aplicar, implicaciones docentes, diseños de materiales, con el objetivo de optimizar la explotación de las TIC en entornos de formación superior.
El Programa de Educación a Distancia, nace como respuesta a las necesidades de articulación e implementación de la modalidad a distancia para la atención de los estudiantes de la región, que desde los orígenes la Institución atiende bajo la denominación de Alumnos No Residentes. Se pretendía entonces sistematizar y organizar los esfuerzos que permitan una gestión eficiente de los recursos puestos al servicio del mejoramiento de la calidad de la enseñanza. En este sentido, una de las acciones concretas que se implementó en el año 2004, fue el Sistema Educativo Unpabimodal, que conjuga acciones pedagógicas, administrativas y de seguimiento y asistencia para aquellos alumnos que no pueden asistir a las sedes de Universidad para realizar sus estudios por cuestiones espacio-temporales.

Metodología utilizada.
La incorporación del Sistema Educativo Bimodal de la UNPA se realizó adaptando la metodología de desarrollo establecida por el Proceso de Plan de Acción de Sistemas[1]. El proceso unificado presenta tres características fundamentales que se consideraron importantes para transitar el modelo de docencia presencial de la Universidad al sistema bimodal, en tanto que: está dirigido por los casos de uso (es decir, qué y para qué se quiere desarrollar), está centrado en la arquitectura (es decir en una representación del sistema que incluye los componentes estructurales, el comportamiento visible de esos componentes para el resto del sistema y el modo en que dichos componentes interactúan) y es iterativo e incremental (lo iterativo se basa en la evolución de prototipos o modelos que se analizan, discuten y ponen en funcionamiento, para ser utilizados con los usuarios y gestores), de estas iteraciones, surge la adopción, actualización o ajuste del modelo anterior, posibilitando acceder a la maduración del sistema definitivo (ver fig. 1). En este sentido se transitaron las siguientes etapas de metodología establecida: a)Iniciación, b) Elaboración del Sistema, c) Construcción y d) Transición (Liberación y Soporte)

El proceso definido por la UNPA en el marco del Programa de Sistemas Y tecnologías de la Información, adhiere al modelo UML. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software entendido como software de gestión organizacional (incluye el manejo de la información y los actores involucrados en el proceso de generación del capital cultural organizativo) y no como una aplicación de software en el sentido común de la expresión

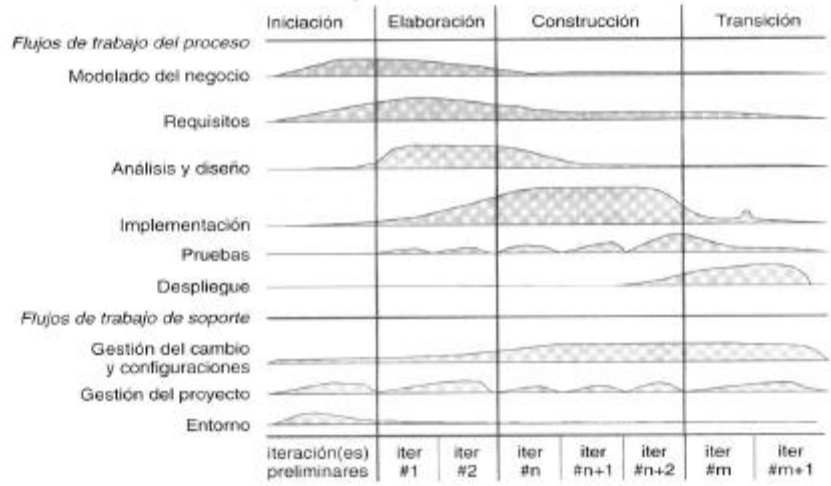


Figura C.1. El ciclo de vida del desarrollo de software.

En la etapa de Iniciación del Proyecto se reconoció formalmente la necesidad de definir un nuevo modelo educativo para la UNPA asignándole capital humano, recursos materiales y financieros, e informando a los involucrados sobre lo definido. El objetivo de la etapa fue la de obtener una visión clara del alcance del sistema, su adherencia a los objetivos de la Institución y a los requerimientos de los gestores institucionales, alumnos, docentes y los miembros de la comunidad a la que sirve.

En esta etapa se definieron las características del modelo pedagógico, asociada a la gratuidad de la formación de grado, a la zona geográfica prioritaria de influencia de las actividades de docencia de grado, la definición de la bimodalidad y la incorporación del profesorado presencial como actores naturales de la bimodalidad, se buscan otras Universidades como "socias" en la gestación y desarrollo del nuevo modelo y los estándares de calidad. Se establecieron como principales riesgos, las tensiones políticas, el rechazo del profesorado, la capacidad de financiamiento sostenible del proyecto. Se definió en consecuencia la priorización de la formación del profesorado, la incorporación de los gestores como aliados en la definición de los presupuestos institucionales y definición de la infraestructura tecnológica, se implementa una política de concursos de proyectos de innovación pedagógica, con subsidios al profesorado.

En la etapa de la Elaboración del Sistema se profundizó el estudio de los procesos de negocios de la institución[2] y los requerimientos de los usuarios, se evaluaron mejoras y se definieron nuevos procesos para la ejecución del trabajo en las áreas afectadas y en base a los mismos se hicieron propuestas de reformulación de normativas vigentes. En este sentido se diseñó la arquitectura del sistema en lo relativo a las funciones asociadas a la incorporación del entorno virtual, y si se analizó en detalle la arquitectura anterior

[2] Entendidos los procesos de negocios, como las reglas y criterios de funcionamiento de los procesos llevados a cabo por la Universidad, los cuales definen el comportamiento y las funcionalidades que deben sostener los sistemas educativos, de acuerdo al modelo pedagógico que desea desarrollar

actualizando ciertas definiciones de articulación y complementación entre sedes, para las funciones de grado, se generó un corpus normativo básico que estructura el sistema bimodal. Se analizaron los requisitos de calidad para la migración de un modelo presencial con algunos esquemas de semipresencialidad, al nuevo modelo educativo. Se considera necesario la búsqueda de cooperación y colaboración entre sedes.

Durante las iteraciones correspondientes a esta etapa aparecen, entre otros, dos elementos importantes que ajustan las definiciones iniciales, el primero asociado a la definición de grados de estandarización de la asistencia técnico pedagógica[3] (SATEP) requerida por el estudiante o definida por el equipo docente, el modelo original sostenía un modelo dicotómico, ajustándose a cinco estándares; el segundo asociado a los requerimientos de bibliotecas y bancos de datos prioritarios, el tercero el ajuste de las responsabilidades de la gestión y seguimiento del desarrollo curricular. Se analizan aquí también los principales riesgos, estableciendo el factor político como una de las principales barreras, el segundo factor la disponibilidad real de la información necesaria por parte del estudiante, la falta de herramientas para la toma de decisiones de los gestores académicos. Se define entonces una estrategia de incorporación activa del estudiante presencial, en la utilización de las herramientas tecnológicas para el soporte y apoyo de las actividades cotidianas, se continúa con el plan de asistencia y formación del profesorado, se potencia el concurso de proyectos de innovación pedagógica, se comienza con el plan de formación de los gestores y con la construcción de herramientas para la administración y generación de información que posibiliten el seguimiento curricular y la toma de decisiones al respecto. Se define el plan de migración y la articulación con los sistemas de información académica. Comienza el problema de la información de base.

Durante la etapa de Construcción (tercera del proceso) se requiere contar con todos los documentos de diseño y arquitectura aprobados por los órganos de gobierno correspondientes, se avanza sustantivamente con el modelo tecnológico, definiendo el modelo de integración de las tecnologías, se inicia el desarrollo de la documentación que se utilizará en las capacitaciones, se confeccionan y ejecutan las pruebas del sistema. Se preparan los productos relacionados a las migraciones que correspondan.

En esta etapa se trabaja fuertemente el apoyo al profesorado, el apoyo a los gestores institucionales, el apoyo y seguimiento de los estudiantes. Se delinean e incorporan sólo parcialmente la estructuras administrativas. Se establece como requisito para la creación de nuevas áreas institucionales, el acuerdo entre sedes para la constitución de áreas en función de la complementación de especialidades sostenidas por diferentes sedes. Se establecen nuevos criterios para los subsidios de los proyectos de innovación pedagógica. Se establece por primera vez, una oferta de grado cuyos espacios deben garantizarse en el entorno virtual, pudiendo además ser implementada en la modalidad presencial. Ninguna sede la implementa presencial.

Aparece como principal riesgo el factor político, las tensiones entre sedes por el potencial crecimiento y por las diferencias de capacidades adquiridas en la bimodlidad de cada una de ellas, se desnudan la falta de liderazgos académicos entre sedes, para algunas

[3] Sistema de asistencia técnico pedagógica del modelo educativo de la UNPA que establece básicamente la definición la relación docente- estudiante, estudiante-estudiante, estudiante-materiales, estrategia comunicacional, grado de presencialidad obligatoria, formas de acreditación. (SATEP: Márquez E.M., Rojas H.S., Bain M.E., Leno, M.J.)

disciplinas, se requiere mayor capacidad de gestión académica. Se constituye en crítico la falta de información de base, los programas[4] de los espacios curriculares, los tiempos de respuestas y la precisión y oportunidad de la asignación de los recursos humanos, de las ofertas de espacios curriculares, de la bibliografía y fuentes documentales consideradas como obligatorias por los docentes, los sistemas de información de los gestores académicos.

En consecuencia se continúa con las instancias de capacitación, sensibilización y difusión de las buenas prácticas y las lecciones aprendidas, se media en las tensiones entre sedes, se sostiene a los diferentes actores involucrados desde dos enfoques, en la generación de las competencias para los que menos tienen y en la potenciación de los que vienen desarrollándose. Los estudiantes son los aliados estratégicos. Se buscan nuevos modelos de asignación de recursos humanos, se comienza con la concientización del valor de la información y con la instalación de una cultura de la bimodalidad en la UNPA.

Por último la etapa de Transición o Liberación. (Ver fig 2).La etapa de Transición es aquella en la cual se deja el sistema operando en modo real y se realizan las acciones necesarias para garantizar que el mismo funcione adecuadamente y no se requiere la intervención excepto para nuevos requerimientos. Se ha realizado la implantación[5] de nuevo Sistema Educativo Bimodal.

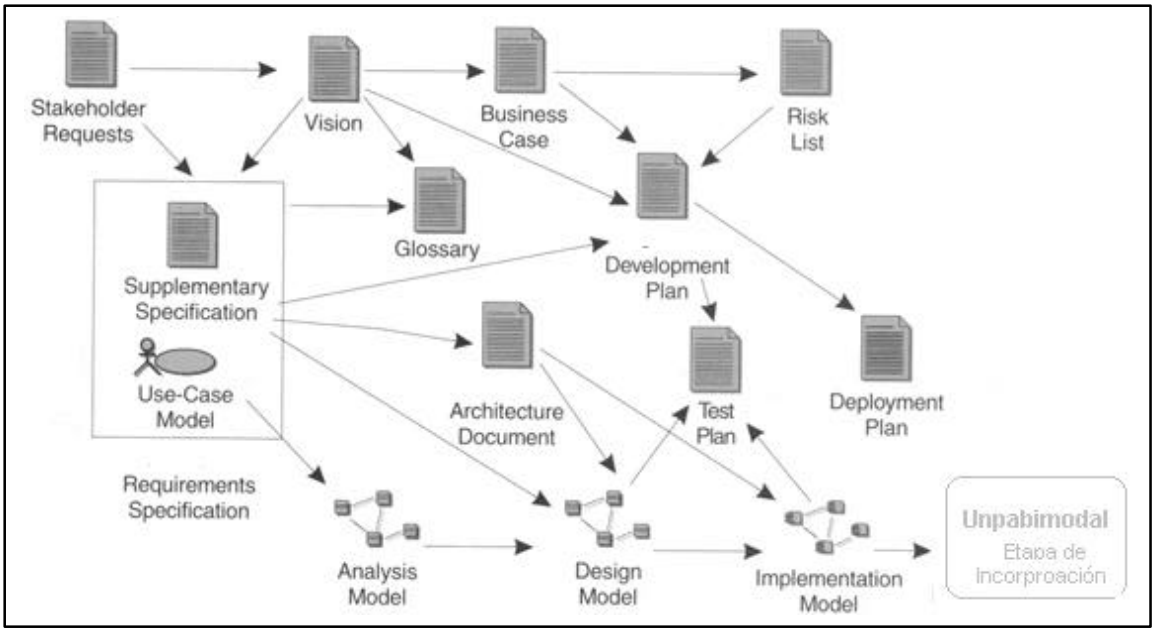


Fig 2.

[4] También denominado plan de la asignatura.

[5] Es decir que se ha establecido una nueva forma de hacer las cosas, comienza la gestación de una nueva cultura para el desarrollo de la docencia de grado. Es el inicio de una nueva etapa. Se ha definido el cómo de ha de hacer para todo aquel que comienza en la UNPA.

Conclusiones

En primer término, consideramos adecuada la adaptación del proceso unificado, como metodología para la definición e implementación de un Modelo Educativo Bimodal, como proceso iterativo y participativo en un proceso investigación y desarrollo (I+D), sostenido en los casos de usos.

En segunda instancia, consideramos que durante dicho proceso, aparecen el conjunto de elementos o componentes que caracterizan o definen los modelos educativos virtuales analizados por Gallardo Pérez, Torrandell Serra y Negre Bennasar. Sin embargo, se consideran de importancia algunos que, si bien pueden ser considerados como parte de otros componentes, por el grado de sensibilidad que le aportan al grado de éxito o fracaso en la cultura organizacional de una nueva forma de impartir la docencia, se considera necesario analizarlos. En este sentido consideramos necesario incorporar a) la capacidad de gestión académica: entendida como la capacidad de asignación y seguimiento del profesorado, b) la intensidad o importancia de la tensiones políticas en las organizaciones universitarias como factor propio de la horizontalidad de su poder, c) la capacidad de liderazgo académico de los grupos disciplinarios y su capacidad de asociatividad d) la capacidad de apoyo al profesorado en la incorporación al nuevo modelo, e) la cohesión y disponibilidad de los servicios de administración y gestión de la información.

En cuanto a los resultados en la UNPA se requiere una redefinición de los mecanismos de seguimiento institucional, una redefinición y seguimiento en las funciones de gestión del profesorado, consolidando el rol de gestor de recursos humamos, una redefinición de los sistemas de información institucional con miras a la integración y a la constitución de sistemas de gestión del capital organizacional sostenido en el capital cultural de la misma, pudiendo arriesgar que si bien la implantación del sistema educativo bimodal ha generado necesidades de información institucional, es también real que ha puesto de manifiesto una falta de información de base que los sistemas presenciales dan por existentes.

Referencias

SALINAS, J, NEGRE, F, GALLARDO A (2006). *Modelos didácticos en entornos virtuales de formación. Identificación y valoración de elementos y relaciones en los diferentes niveles de Gestión*. EDUTEC'06. La educación en Entornos Virtuales: Calidad y efectividad en el E-Learning. (Tarragona, España).

ROJAS, Hugo, E.MÁRQUEZ, I.ARANCIAGA, S.RAMOS, M.. BAIN. *Edudiseños, Tecnodesigños u Orgadestinos*. IV Seminario Internacional de Educación a Distancia. Córdoba, Argentina. 2006.

GALLARDO, A., TORRANDELL, I, NEGRE, F. (2005): *Estudio de modelos organizativos en la enseñanza universitaria mediante entornos virtuales*. EDUTEC '05. Congreso Internacional sobre Formación del profesorado y Nuevas Tecnologías. Santo Domingo (República Dominicana).

BARBERÁ Elena. *La incógnita de la Educación a Distancia* . ICE-HORSORI, Primera Edición, 2001.

García Aretio, Lorenzo. *La educación a Distancia: De la teoría a la práctica*. Barcelona. Ariel Educación, 2001.

BATES, A. (2000): *Como gestionar el cambio tecnológico*. Barcelona, España Editorial Gedisa, 2000.

RUMBAUGH, James, Ivar Jacobson, Grady Booch. *El lenguaje unificado de modelado*. Madrid: Addison Wesley, 1999.

MARTINEZ SANCHEZ, Francisco (Comp). *Redes de comunicación en la Enseñanza*. Madrid. Ediciones Paidós, 2003.

VALHONDO, Domingo. *Gestión del Conocimiento.* España. Ediciones Diaz de Santos, 2003.

ARBONIES, Angel. *Conocimiento para innovar*. España. Ediciones Diaz de Santos, 2006.

Propuesta metodológica para el desarrollo de materiales de estudio hipermediales para la articulación Escuela Media y Universidad[1]

Alejandro Héctor Gonzalez[2]
agonzalez@lidi.info.unlp.edu.ar

Instituto de Investigación en Informática (III-LIDI) – Facultad de Informática – UNLP

Abstract

This paper presents a methodological proposal for the development of hypermedia resources from the educational point of view. The characteristics of entering students and their articulation within High Schools are analyzed. The case experienced in the School of Computer Sciences is presented together with a methodology of designing hypermedia material oriented to this student's type, focusing on the cognitive skills necessary to solve problems.

For the presentation of the proposal, the characteristics of instructional design models and the involved stages are analyzed. During the building process of the material, the information is treated in hypertextual and multimedia format so as to position the student in the heart of the educational process. The concept of learning objects for the realization of the material is also introduced together with some of the elements to be taken into account in designing hypermedia resources: didactic design, narrative elements, and used digital media.

Keywords: *Methodology, Multimedia, Interactivity, Learning Objects, Instructional Design.*

Resumen

En este trabajo se presenta una propuesta metodológica para el desarrollo de recursos hipermediales desde el punto de vista educativo. Se analizan las características de los alumnos ingresantes y su articulación con la Escuela Media. Se presenta el caso de la Facultad de Informática y se propone una metodología de diseño de materiales de estudio hipermedia orientada a este tipo de alumnos, poniendo atención a las habilidades cognitivas necesarias para la resolución de problemas.

Para la presentación de la propuesta se analizan las características de los modelos de diseño instruccional y las etapas involucradas. Dentro del proceso de construcción del material se tiene en cuenta el tratamiento de la información con formato hipertextual y multimedia de manera de posicionar al alumno en el centro del proceso educativo. Se introduce el concepto de objetos de aprendizaje para la realización del material y algunos de los elementos a tener en cuenta en el diseño de los recursos hipermedia: diseño didáctico, elementos narrativos y medios digitales utilizados.

Palabras clave:
Metodología, Multimedia, Interactividad, Objetos de Aprendizaje, Diseño Instruccional.

1 Trabajo dirigido por el Ing. Armando E. De Giusti degiusti@lidi.info.unlp.edu.ar.

2 Jefe de Trabajos Prácticos DE – Facultad de Informática – UNLP

1. INTRODUCCION

Los avances en tecnología de la información y comunicación digital de los últimos tiempos ofrecen una variedad de medios para lograr la transmisión de la información. Se caracterizan por los rasgos de inmaterialidad, interactividad, instantaneidad, innovación, interrelación de imagen y sonido, digitalización, mayor influencia relativa de los procesos que de los productos, interconexión y diversidad [20]. Las tecnologías de la información y la comunicación (TIC's) han cambiado la forma de comunicación de las personas y en este contexto la Educación no es ajena a tales modificaciones.

En este contexto nos situamos ante un nuevo sujeto que aprende y que requiere de los medios necesarios para poder llevar adelante el proceso de aprendizaje. Los alumnos que llegan hoy a la Universidad han pasado por diferentes experiencias de uso de tecnologías digitales que van desde el celular hasta simulaciones en la Web. Los materiales de estudio deben ser acordes a los trabajados por los estudiantes, debe proveerse la "familiaridad" necesaria para lograr un proceso de articulación adecuado para cada alumno, de manera que no se presente el modo universitario como extraño y ajeno a la realidad del estudiante

Se presentan, además del motivo tecnológico, diversos factores que influyen en la inserción a vida universitaria. En las carreras universitarias en Argentina y en particular las referidas a Informática existe un fenómeno significativo referido a la deserción en los primeros años de estudios. Esto muestra un problema en la transición entre la Escuela Media y el inicio de la carrera universitaria que denota una falta de adecuada articulación entre ambos sistemas [22]. En general las instituciones académicas tratan de establecer acciones/programas tendientes a disminuir este desgranamiento inicial a través de propuestas educativas innovadoras que incluyen el empleo de TICs [18].

En la Facultad de Informática de la UNLP se trabaja una propuesta educativa enfocada a la retención de los alumnos en las etapas tempranas de su carrera, considerando especialmente la reducción de las diferencias entre la Escuela Media y la Universidad. En este contexto, en este trabajo se presenta una propuesta metodológica de desarrollo de materiales hipermedia dedicados a acciones de articulación con la Escuela Media, para mejorar la información y preparación específica de los alumnos y facilitarles el acceso y permanencia en la Universidad [26][27][28][29].

2. ETAPAS DE LOS MODELOS DE DISEÑO

El Diseño instruccional como teoría ofrece una guía que indica cómo ayudar a las personas a aprender y desarrollarse mejor. Es el intento por relacionar eventos de instrucción específicos relacionados con el proceso de aprendizaje y los resultados a obtener al final de la instrucción [4].

Según Reigeluth, las principales características del diseño instruccional como teoría son:

- Como es "orientada hacia el diseño", se concentra en los medios que permiten la obtención de los objetivos de aprendizaje y desarrollo, de modo que resulte práctico y útil para los educadores, que deben poder mostrar cómo lograr sus metas u objetivos de aprendizaje.

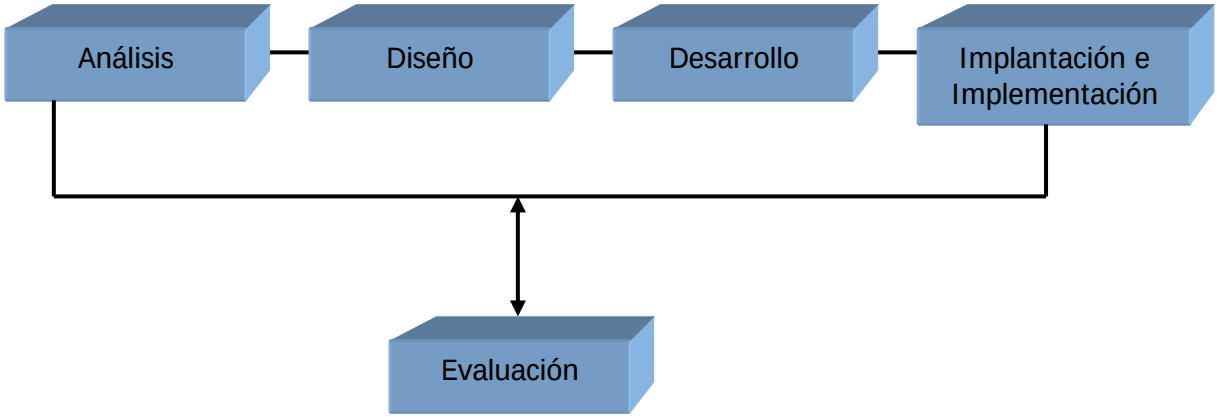
- Es prescripta (describe cómo es el ambiente de aprendizaje y si se puede construir o alterar), dado que ofrecen los lineamientos para realizar las acciones que nos conduzcan hacia el logro de ciertos resultados.

- Debe identificar métodos de instrucción y situaciones en las que se puedan utilizar. De esta forma los métodos son situacionales y no universales. A su vez los métodos de instrucción se pueden dividir en componentes más detallados que proporcionen más lineamientos para los educadores.

- Los métodos se consideran más probabilísticos que determinísticos pues incrementan las posibilidades de lograr las metas. Una meta desde el punto de vista de la teoría de diseño instruccional es obtener mayores posibilidades para propiciar que los resultados deseados ocurran. y tiene un valor o una filosofía que lo soporta. Los valores son primordiales al decidir que vías se han de seleccionar en cuanto al método para obtener esas metas.

El diseño instruccional como modelo es la representación visual de un proceso de diseño instruccional que muestra los elementos principales o fases y la relación que cada uno guarda entre ellos [5]. Según Gros los modelos de diseño instruccional tratan de establecer un puente entre las teorías de aprendizaje y la práctica al construir un sistema instruccional.

El diseño instrucional esta formado por etapas que proveen el esquema sobre el cual se produce la instrucción de forma sistemática. Las fases o etapas en general pueden resumirse en: Análisis, Diseño, Desarrollo, Implantación y Evaluación [6], como se muestra en la siguiente figura:



- En la etapa de *análisis* se define el problema, se identifica la fuente del problema y se determinan las posibles soluciones.

- En la fase de *diseño* se utilizan los resultados de la fase de análisis para planificar una estrategia y así producir la instrucción. Se realiza un bosquejo de cómo alcanzar las metas instruccionales.

- En el *desarrollo* se construyen los materiales que se van a utilizar. En esta etapa se elabora la instrucción, los medios que se utilizarán en la instrucción y cualquier otro material necesario, tal como guías de aprendizaje.

- En la *implantación e implementación* se pone en práctica la instrucción. La misma puede ser llevada a cabo en diferentes ambientes, dependiendo de la estrategia diseñada: en el salón de clases, en laboratorios o en entornos Web de enseñanza y aprendizaje. En esta fase se

propicia la comprensión del material, el dominio de destrezas y objetivos, y la transferencia de conocimiento del ambiente instruccional al ambiente de trabajo.

- La fase de *evaluación* atraviesa todo el modelo y debe darse en todas las etapas del modelo instruccional; y se evalúa la efectividad y eficiencia de la instrucción.

El diseño instruccional está íntimamente relacionado con las teorías de aprendizaje y a su vez, esto tiene una representación o modelo. Algunos modelos como el de Dick & Carey, (1990, 1996) basado en principios de Gagné prescribe un diseño que va descomponiendo los elementos de la instrucción a componentes más pequeños y trata de observar los resultados del aprendizaje sobre la base de ciertas variables de entrada y se lo considera de características conductistas. En cambio el modelo de Jerold & Kemp (1985) es un modelo centrado en las necesidades del estudiante, las metas, prioridades y las limitaciones sustentados por la evaluación formativa y sumativa y en cada fase debe haber una revisión , se encarga de observar los procesos de construcción del conocimiento que ocurren en el estudiante y se lo considera de base constructivista [9][10].

También están los modelos orientados a la cognición distribuida que, además de considerar los procesos cognitivos y constructivos del aprendizaje, incluyen los conceptos de mediación y el aprendizaje social de Vygotsky [14]. En este modelo el aprendizaje es el resultado de la interacción del individuo en el mundo (medio ambiente, cultura, interacción social). No sólo considera los procesos cognitivos y de construcción de conocimiento del usuario en forma individual, sino que integra la interacción con grupos y/o tutor para obtener un modelo integral de educación. Motiva el uso de diagramas para visualizar la estructura y el contenido, se basa en la información disponible y motiva el trabajo en equipo, tendiendo hacia un aprendizaje colaborativo y cooperativo [9].

Los modelos de diseño instruccional se pueden utilizar para producir el desarrollo de una clase con los temas a abordar, los módulos para lecciones, un curso de una carrera universitaria, cursos de diversa índole, etc.

3. OBJETOS DE APRENDIZAJE

Según Chan el diseño instruccional tiene ante sí el reto de generar un objeto para el estudiante, a partir de la información y la manera de presentarla, de manera que le resulte una herramienta para aprender. El concepto de objetos de aprendizaje se ha presentado en los últimos años como respuesta a la necesidad de diseñar procesos de enseñanza y aprendizaje desde un enfoque constructivista, ayudados por al tecnologías digitales [11].

David Willey considera que un objeto de aprendizaje (OA) es un recurso digital pensado para apoyar los procesos de aprendizaje y con posibilidad de “reuso”. Los describe como pequeños componentes instruccionales (‘chunks’), re-usados un número de veces en diferentes contextos de aprendizaje y en general son entidades digitales *re-usables, adaptables, escalables, con capacidad generativa y distribuidos por Internet.* Podemos entender un OA como un material con diseño educativo delimitado, que propicia la actividad interactiva y está diseñado para facilitar el aprendizaje.

Wiley hace referencia explícitamente que el OA debe ser digital, punto que no es acordado por todos los autores que actualmente consideran que un objeto “no digital” también es un OA. y que este concepto no es nuevo sino que siempre se ha trabajado con la idea de “reutilización”, “adaptabilidad” y “escalabilidad”.

También se introduce el concepto de granularidad de los OA, que se refiere al grado de detalle o precisión en tamaño, potencialidad de reutilización y posibilidad de descomposición de los mismos.

Otra característica es su independencia del espacio virtual donde se encuentre almacenado para su distribución (Repositorios de OA). Un OA debe estar "etiquetado" para poder localizarlo, buscarlo o seleccionarlo (metadato). Los metadatos se refieren a información semántica asociada al contenido del aprendizaje que no forman parte de éste como facilidad de acceso y recuperación, estilo didáctico, formato y destinatarios. Ayudan a seleccionar OA construidos por otros [14] [15].

Un OA no es en absoluto del todo reutilizable en cualquier situación, sino que requiere ser integrado dentro de una secuencia de aprendizaje previamente diseñada y basada en una teoría concreta [23]. A su vez el ámbito educativo, las condiciones sociales y culturales, los diferentes estilos de aprendizaje, las motivaciones y otros tantos aspectos referidos a docentes y alumnos tendrán que ser tenidos en cuenta en la definición y etiquetado de los objetos de aprendizaje para que éstos puedan responder realmente a diferentes necesidades y usos [12].

4. METODOLOGÍA PROPUESTA

Teniendo en cuenta las fases de diseño instruccional y tomando los modelos basados en cognición distribuida se propone una metodología de trabajo para el armado de materiales hipermedia para alumnos ingresantes a la Universidad y en particular se aportan sugerencias para el caso de los alumnos de informática.

La metodología esta pensada para realizar un trabajo de desarrollo en equipo. En el caso de no contar con el profesional indicado para cada etapa, una misma persona debe cumplir diferentes roles. En cada fase (análisis, diseño, desarrollo, implementación y evaluación) de la propuesta se indican las tareas generales, los roles necesarios y las particularidades en el caso de los alumnos de informática

Análisis: en esta etapa deben definirse los elementos que forman el marco general del material hipermedia a desarrollar. Se debe tener en cuenta las características de los alumnos, a quien va dirigido, llevar adelante un relevamiento del contenido actual y sus formatos: textos, páginas Web, apuntes, animaciones, etc. Además debe realizarse un análisis del contenido educativo y de las teorías de aprendizaje que subyacen en cada caso.
Para este relevamiento se deben realizar encuestas y entrevistas con los docentes y alumnos para poder identificar posibles fallas en el desarrollo de un tema o contenido en particular. También se sugiere realizar una búsqueda en la Web acerca de los desarrollos que existan al momento y que estén referidos al contenido hipermedia a desarrollar de manera que resulte de fuente de información para diseño.

Es importante en esta etapa poder determinar el tipo de material a desarrollar [24]:

Tutoriales: El programa actúa como un "usuario experto" que va guiando al estudiante en el recorrido de los contenidos.

De ejercitación: El programa se basa en la "resolución de ejercicios y problemas", más que en la presentación de conceptos nuevos. Ante cada acción del estudiante, el programa le presenta Feedback inmediato, que lo ayuda a mejorar sus prácticas.

De simulación: El programa presenta al estudiante situaciones similares a las de la realidad. En cada una de ellas, el estudiante puede actuar "como si" estuviera actuando en la realidad.

Hipertextuales: El programa se basa en la relación de conceptos a través del "hipertexto" como herramienta.

Referenciales: Programas donde es posible "buscar información".

En el caso de los alumnos ingresantes a informática se recomienda indagar sobre el nivel de conocimiento previo en computación: manejo de Pc, reparación, programación, utilización de Internet y manejo de aplicaciones en general, tipo de educación recibida en informática (escuela media, Instituto, curso, etc). En el caso de los materiales de estudio deben revisarse si en los objetivos del aprendizaje se hace referencia al desarrollo de habilidades cognitivas de orden superior y que mecanismos se ofrecen para poder alcanzarlas.

Se necesita para esta etapa un experto en contenidos que se encargue de revisar la secuenciación de los contenidos, y también de un experto en diseño en materiales de estudio en diversos formatos. El resultado es la obtención de un documento donde aparezcan las metas instruccionales y una lista de las tareas/contenidos a que van a ser necesarios para el aprendizaje. Se debe obtener una definición de la necesidad educativa a través del relevamiento de información realizado y contar con un brief del material a diseñar. El documento debe incluir: el título del proyecto, los objetivos, destinatarios, los nombres de los responsables, la fecha de inicio y el tiempo estimado de producción, el soporte del material a desarrollar (cd, web, video, etc) y la plataforma digital sobre al cual ejecuta.

Diseño: en esta etapa se trabaja el detalle del material. Básicamente se debe elabora el diseño de la interfase y el guión multimedia. En el diseño de la interfase se debe tener en cuenta una metáfora comunicacional que refleje el tema a tratar, debe incluir el esquema de contenidos y el recorrido de la obra multimedia obtenido a partir de un diagrama de navegación previo.

El guión multimedia en cambio, es la descripción pormenorizada de cada una de las escenas de cada pantalla de la cual contará la obra. Aquí se vuelcan las estrategias didácticas, la presentación de los contenidos, las estrategias de interactividad. Se seleccionan para cada caso los medios a utilizar y la fuente de origen: texto, imagen, video, sonido, animación, etc.

Teniendo en cuenta a Bou Bouzá [1] se deben determinar las generalidades del diseño teniendo en cuenta:

- la interactividad (acción o dinámica que ocurre entre la persona y el medio);
- el principio de libertad (como el usuario puede navegar la obra, los circuitos posibles);
- el principio de vitalidad (cuan dinámica es a la vista la pantalla);
- la necesidad de la obra (diseñarse teniendo en cuenta la necesidad educativa planteada);
- el principio de atención (el receptor debe mantener la expectativa ante el recorrido de la aplicación, esta debe ser cognitiva (con información relevante y bien organizada) y afectiva (tiene que ver con el lazo afectivo que se pueda lograr con el usuario, su identificación);
- el principio multicanal (mensajes enviados por diferentes medios como : audio, video, imagen ,etc);
- las múltiples entradas (deben ser adecuadas de acuerdo a la capacidad cognitiva del alumno).

Debe también en esta etapa, definirse el modelo cognitivo especificando el tratamiento del contenido, su organización y secuencia. Se sugieren desarrollo de mapas conceptuales para

favorecer la relación de conceptos y tender a un aprendizaje significativo [25]. Construir actividades dependiendo del esfuerzo cognitivo requerido para el alumno. No debe perderse de vista la relación de la práctica con la teoría. Proponer actividades optativas, obligatorias, equivalentes. Se sugiere diseñar actividades que promuevan el trabajo en equipos o grupos de trabajo, para llevar a cabo la interacción social.

Atendiendo a las particularidades de los alumnos ingresantes se sugieren actividades que trabajen la resolución de problemas, análisis de textos en general (la lectura de enunciados, identificación de datos relevantes e irrelevantes para un problema),con un tratamiento del error adecuado y con el feedback necesario, de manera de lograr [25]:

- **Reconocimiento**: a través de ejemplos de solución de problemas similares.
- **Comprensión:** el alumno debe poder reproducir la información, teniendo en cuenta su estructura semántica y su significado.
- **Reconstrucción global:** el alumno debe usar la información que se le presentó en la resolución de un problema, para lo que necesita reconstruir su estructura semántica.
- **Interpretación constructiva**: plantearse nuevos problemas a partir de la información que tiene sobre una determinada área de conocimiento.
- **Ayudas o "pistas"** para ayudar al alumno a encontrar la respuesta correcta.

En cuanto al *tratamiento del error* respecto de la respuesta correcta se debe ofrecer un feedback sumativo o formativo (explicativo). Se sugiere el *uso del conflicto* de manera de lograr favorecer la discusión y la confrontación de ideas.

Otro elemento a tener en cuenta es el *elemento narrativo* del diseño (discurso). Este discurso hace referencia a la forma en que aparecen y se suceden todas las imágenes, textos, sonidos y demás elementos en los que se apoya el mensaje [1].

Por ejemplo: texto, hipertexto, principio de coherencia y sorpresa-coherencia, principio de economía (la narración no debe ser redundante, el discurso debe ser fluido y se debe tener en cuenta que el receptor del mensaje es rápido en sus estrategias cognitivas de lo que el guionista supone) referida al tiempo, espacio, conceptos, lenguaje y espera. Estos elementos narrativos deben estar presentes en la obra de manera de generar el impacto ("o residuo") cognitivo deseado.

En este punto es importante considerar la incorporación de personajes que acompañen al estudiante durante el proceso de aprendizaje. El buen desarrollo de personajes y una trama producen, según Bouzá, un acompañamiento en el proceso de aprendizaje que favorece la familiaridad con el recurso y el seguimiento de los diferentes recorridos para ver que ocurre. Desde este punto de vista el diseño instruccional puede tomar esta característica y reorganizar los contenidos de manera que el alumno, acompañado de los personajes con características tutoriales, recorra los temas y los incorpore a su proceso de aprendizaje.

El discurso es un elemento del diseño muy importante en el caso de ingreso a la universidad, dado que los alumnos provienen de diferentes escuelas con diferentes estilos narrativos y con diferentes

experiencias vividas. Se sugiere analizar la incorporación de personajes de manera de articular los diferentes casos, acercado así a un lenguaje común.
Los roles involucrados en esta etapa corresponden en su mayoría a un diseñador gráfico, y un diseñador instruccional. Esta etapa produce básicamente un documento que es el guión multimedia que sirve de apoyo a la construcción del material en el cual debe tenerse en cuenta la interactividad de manera de lograr recursos que den la posibilidad de elección de diferentes recorridos en el contenido, disponer de actividades que pongan en juego diferentes estrategias cognitivas y metacognitivas de resolución de problemas, acompañadas con el Feedback necesario.

Desarrollo: en esta etapa se desarrollan todos los materiales necesarios para llevar adelante la instrucción. Dependiendo del tipo de material se elegirán los programas más adecuados para lograr la integración de todos lo medios elegidos.

En el caso de páginas Web se debe elegir un software que facilite el trabajo. Si no se dispone de programadores y el docente debe llevar adelante esta tarea debe pensar en software, por ejemplo tipo eXe, que permita generar una interfase estándar rápidamente, la cual pueda luego adaptar al diseño gráfico e isntrucccional elegido. La propuesta contempla generar objetos de aprendizaje los cuales deben atender a estándares predefinidos como SCORM.

En esta etapa toman relevancia los roles del programador y del diseñador gráfico. Los objetos de aprendizaje que se obtienen como resultado de esta fase deben, luego de haberlos probado y evaluado, ser ubicados en un repositorio de aprendizaje para compartirlo y poder ser re-utilizado por otros docentes.

Implantación: está ligada a la propuesta educativa en sí misma: se deben tener en cuenta las condiciones de trabajo tanto de alumnos como de docentes y definir el entorno de trabajo ya sea en modalidad presencial, semi-presencial o a distancia.

Si bien el material se diseña para que un alumno recorra el contenido de acuerdo a sus necesidades, se debe pensar en una actividad integradora compartida con otros estudiantes de manera de fomentar el trabajo grupal y discusión en grupos, asistidos por un coordinador o tutor. Por ejemplo si el material es Web y provee un trabajo en foros debe pensarse en la participación del tutor como coordinador de los temas. Se puede utilizar parte de la clase para discutir sobre un determinado recorrido del material y compartir los resultados obtenidos, etc.

Deben atenderse en este caso a las variables involucradas en los ingresantes, teniendo en cuenta sus diferentes estilos de aprendizaje de manera de articularlos y enseñar a utilizar las habilidades cognitivas de orden superior tales como la metacognición y la abstracción.

Los roles de esta etapa son esencialmente docentes como guía con un estilo tutorial y el resultado es la puesta en práctica y los resultados de aprendizajes logrados por cada alumno.

Evaluación: esta etapa se lleva a cabo durante todo el proceso de construcción del material y requiere una revisión y evaluación del cada resultado de cada etapa y de todo el proceso.

En especial se debe realizar una evaluación de la aplicación del producto para revisar las fortalezas y debilidades encontradas en la implementación. Se puede realizar a través de cuestionarios, listas control entrevistas, etc. La evaluación es de tipo formativa y sumativa. Se la considera “formativa” porque se desarrolla en forma es continua, se lleva a cabo mientras se están desarrollando las demás fases. El objetivo de este tipo de evaluación es mejorar la instrucción antes de que llegue a la etapa

final. En cambio la evaluación "sumativa" se da cuando se ha implantado la versión final de la instrucción. En este tipo de evaluación se verifica la efectividad total de la instrucción y los hallazgos se utilizan para tomar una decisión final, tal como continuar con un proyecto educativo o comprar materiales instruccionales. También incluye esta etapa la planificación de evaluaciones estudiantiles del curso para mantener al tutor consciente de las necesidades y demandas.

En esta etapa se involucran los roles académicos (docentes, tutores, profesores) y de diseño instruccional. Se producen los documentos de evaluación y estadísticas para ver el avance y resultados del material.

5. TRABAJO EN DESARROLLO

Se está trabajando en un material hipermedia de tipo tutorial con ejercitación para los alumnos ingresantes a las carreras de Licenciatura en Informática y Licenciatura en Sistemas de la Facultad de Informática de la Universidad Nacional de La Plata. El curso de ingreso esta formado por tres asignaturas: EPA (Expresión de Problemas y Algoritmos), COC (Conceptos de Organización de Computadoras) y MAT (Matemáticas 0).

Se elige la asignatura EPA la cual introduce al alumno en el desarrollo de algoritmos a través de un lenguaje de programación con un conjunto de instrucciones reducido, denominado Visual DaVinci. Existe un largo trabajo de desarrollo educativo en este módulo, conducido por los docentes de la asignatura inicial de Algorítmica de las carreras de la Facultad [26], [27], [28].

En esta asignatura el alumno debe poner en práctica diferentes habilidades cognitivas para desarrollar problemas, y habitualmente se encuentran diversas dificultades en el proceso de aprendizaje, las cuales llevan a re-ver y re-elaborar habitualmente las prácticas de enseñanza llevadas a cabo.

Se presenta a continuación las etapas desarrolladas al momento de acuerdo a la metodología plateada anteriormente.

Etapa de Análisis: se realizó un relevamiento de las características de los alumnos a través de una encuesta y se tuvieron en cuenta los factores educativos, tecnológicos, sociales y afectivos que influyen en el desempeño de los alumnos durante su primer año de actividad universitaria [18].

Se dispone de información sobre los inconvenientes más comunes que encuentran los alumnos al desarrollar las actividades. Dicha información es obtenida por medio de encuestas, entrevistas con los alumnos, y registro de errores por parte de los auxiliares docentes presenciales.

Se realizó un estudio de los errores cometidos en las pruebas nivelatorias, donde se identificaron los problemas más comunes en la resolución de problemas.

En el ingreso 2007 se realizó una encuesta con una muestra de 150 alumnos para corroborar algunos datos que se presentaban año tras año. Los alumnos manifiestan que en el 60% de los casos comprenden los temas explicados en las clases presenciales tanto teóricas como prácticas. Un 20 % manifiesta entender los contenidos teóricos pero le cuesta comprender la práctica y llevarla al día., el 20 % restante le cuesta ambos contenidos teóricos y prácticos. El 90% manifiesta lograr una integración de todos los temas vistos.

Se identifican los problemas más comunes en la interpretación del enunciado de los problemas planteados y en el contenido referido a modularización y pasaje de parámetros.

Se decide trabajar sobre este punto que es el que la mayoría reconoce tener inconvenientes, de manera de favorecer las estrategias de aprendizaje a través de un material hipermedia de características tutoriales y de ejercitación.

Sumado a esto se proyecta para su implementación llevarla acabo en el ingreso 2008 y tomar un grupo de alumnos para realizar una experiencia piloto. El grupo testigo estará formado por alumnos que utilizarán el material hipermedia y se llevará a cabo una actividad de taller para fomentar el trabajo grupal. Este taller denominado "Taller multimedia" se utilizará para realizar una puesta en común de los temas que ofrece el material y poder tomar una encuesta de opinión para analizar nuevamente la integración de los contenidos.

Se trabajará también con un grupo control que no tendrá acceso al material y servirá para evaluar si no hay otro tipo de modificaciones que afecten la validez del material hipermedia.

Ambos grupos deben ser del mismo turno a cargo del mismo grupo de docentes.

Diseño del material: se tendrán en cuenta las habilidades cognitivas para la resolución de problemas y las características del razonamiento. Para poder presentar los pasos de razonamiento antes diferentes situaciones conflictivas se empelaran personajes que tendrán el rol de acompañar al alumno y achicar la brecha ente el conocimiento experto y el novato. Se intenta atender a las necesidades de articulación entre la Escuela Media y la Universidad, por ello se presentarán personajes que acompañen desde diferentes áreas de estudio: historia, informática, filosofía y matemática.

El material utilizará el estándar SCORM y formato HTML. Los alumnos lo acceden vía Web para hacer el recorrido y realizar las actividades planteadas.

6. CONCLUSIONES Y TRABAJO FUTURO

Se propuso una metodología de trabajo para diseño de materiales hipermedia que intenta favorecer y orientar sobre la construcción de los objetos de aprendizaje, teniendo en cuenta la articulación escuela Media y Universidad.

En el trabajo a futuro se continuará con el diseño del guión y ha comenzado la etapa de desarrollo. Se están perfeccionando los personajes de manera que favorezcan adecuadamente la transición del "principiante" en un tema hasta lograr el conocimiento y manejo del mismo. Se tomará en el elemento discursivo de cada uno de los personajes las habilidades cognitivas encontradas en personajes famosos, referidos a una temática específica, de manera que identifiquen ciertos rasgos de experticia en el contenido tratado.

También se están diseñando los instrumentos de evaluación, para utilizar antes, durante y después de utilizar el recurso.

7. BIBLIOGRAFIA

[1]Bou Bouzá, G. (1997). El guión multimedia. Madrid: ANAYA. Capítulos 1 a 3 y 5.
[2]Moya, Gonzalez Propuesta de desarrollo de material hipermedia para la enseñanza de la Matemática. Presentado en TEYET 2006.
[3]http://www.spsu.edu/htc/hughes/papers/interface.htm
[4]Charles M. Reigeluth M. ed. (1999) Instrucctional Design Theories and Models, A New Paradigm of Instructional, V.II Laurence Erlbaum Associates, New Jersey London, p. 5. apud. Steven J. McGriff, Portafolio, ISD Knowledge Base/ "Theoretical" Introduction en http://www.personal.psu.edu/faculty/s/j/sjm256/portfolio/kbase/Theories&Models/theoryintro.html, recuperado, Junio, 2007. Peter de Lisle (1997). What is Instructional Design? En http://www.coe.uh.edu/courses/cuin6373/whatisid.html y Definitions of Instructional Design, op.cit.
[5]Gros, B., et, al. (1997). "Instructional Design and the Authoring of Multimedia and Hypermedia Systems: Do as Marriage Make Sense?" en Educational Technology, (37), p.48-56, apud. Peter Lisle (1997). What is Instructional Design Theory? en http://hagar.up.ac.za/catts/learner/peterd1/ID%20Theory.htm recuperado Junio, 2007.
[6]The Addie Instructional Design Model en http://distance-ed.fullerton.edu/pages/faculty_online_guide/guide24.htm
[7]Christine M. Remley (2002). Instructional Design Models, en http://www.personal.psu.edu/users/c/m/cmr226/Instructional%20Design/ISD_2.htm recuperado, diciembre, 2003; Peter de Lisle (1997)General IT Topics en http://www.ittheory.com/qual/prep6.htm, recuperado marzo 2005; (s.f.) Models for Instructional Design en http://curry.edschool.virginia.edu/go/ID/ID00-01/models_handout.htm, recuperado noviembre, 2003; Steve Mc. Griff (2001). Instructional Systems Design Models en
[8]Charles M. Reigeluth M. ed. (1999) Instrucctional Design Theories and Models, A New Paradigm of Instructional, V.II Laurence Erlbaum Associates, New Jersey London
[9]La elaboración de un módulo instruccional© Preparado para el Centro de Competencias de la Comunicación Universidad de Puerto Rico en Humacao Proyecto de Título V Gloria J. Yukavetsky, M.A.Ed.
[10]Steve Mc. Griff (2001). Instructional Systems Design Models en http://www.personal.psu.edu/faculty/s/j/sjm256/portfolio/kbase/IDD/ISDModels.html#addie, recuperado marzo, 2005. (Lisle, 1997, op.cit.).
[11]Gisbert, M., Salinas. J., Chan M. E., Guàrdia, L. (2003) Conceptualización de materiales multimedia en Stephenson, J. et al. Fundamentos del diseño instruccional con e-learning. Barcelona: EDIUOC Polsani, P. (2003) Use and abuse of learning objects Learning. Technology Center, University of Arizona, USA. Disponible electrónicamente en: http://jodi.ecs.soton.ac.uk/Articles/v03/i04/Polsani/
[12]Downes, S. (2003). Design standards and reusability. Disponible electrónicamente en http://www.downes.ca/cgibin/website/view.cgi?dbs=Article&key=1059622263&format=full
Gynn, C., Acker, S. (Eds.) (2003) Learning Objects: context and connections, The Ohio State University, USA. Disponoble electrónicamente en: http://morty.uts.ohio-state.edu/learning_objects/documents/TELR-LO7screen.pdf
[13]Martínez, M. (2000). Designing learning objects to mass customize and personalize learning. En The Instructional Use of Learning Objects, Association for Instructional Technology. Disponible electrónicamente en http://www.ait.net
[14]Solomon G, y Perkins , D (2000) Cogniciones distribuidas. Amorrortu, Bs. As.
[15]Koohang, A. and Harman, K. Learning Objects and Instructional Design (Paperback).

[16]Navarro Cedejas , J., Ramirez Anaya L. Objetos de Aprendizaje. Formación de autores con el modelo redes de objetos. UDGVirtual. Colección Cuadrenos de innovación Educativa. México (2005).
[17]Marques Graells, P. (2000). Elaboración de materiales formativos multimedia. Criterios de calidad. Actas del XII Congreso Nacional I Iberoamericano de Pedagogía, vol-II, pp. 288-289. SEP. Madrid, 26-30 septiembre" Madrid: SEP
[18]Gonzalez, Madoz, Gorga. Herramientas tecnológicas de Educación a Distancia en el proceso de articulación Escuela Media-Universidad: el caso de Informática
Reunión: XII Congreso Argentino de Ciencias de la Computación (CACIC 2006). Lugar: San Luis. Argentina.
[19]http://www.me.gov.ar/spu/Publicaciones/publicaciones.html SPU Consultada Julio de2007
[20]Cabero, J. (1996). Navegando construyendo: la utilización de los hipertextos en la enseñanza. En Cabero, J. y otros (coords.): Medios de comunicación, recursos y materiales para la mejora educativa II, Sevilla, CMIDE del Excmo. Ayuntamiento de Sevilla y SAV de la Universidad de Sevilla, 227-243.
[21]Cabero, J (2001) Las TICs: una conciencia global en la educación. En VV.A.A., Ticemur. Jornadas Nacionales TIC y Educación. Murcia:CEP de Lorca, XIX-XXXXVI.
[22]De Giusti, Madoz, Gorga. Análisis del proceso de articulación para Alumnos de Informática, utilizando herramientas de Educación a Distancia. Trabajo aceptado en TE&ET. La Plata. 2006.
[23]García Aretio, L. De la educación a distancia a la educación virtual. Editorial Ariel 2007, capítulo 5.
[24]Bartolome, A. (1994) Sistemas multimediales, en Sancho,J.(coord.)81994). Para una tecnología educativa. Ed. Horsori.
[25] Ausubel, D., Novak, J. y otros: Psicología educacional. Un punto de vista cognoscitivo.
México, Trillas, 1989, cap. 1, págs. 17 a 33.
[26]Russo, Sanz, Madoz, Gorga, Gonzalez. Una estrategia de articulación con el nivel medio: Curso de Pre Ingreso a distancia. Jornadas UNLP 2006.
[27]Madoz , Gorga, De Giusti. Análisis de la utilización de Web Based Learning en un primer curso de Algorítmica. CACIC 2005.
[28]Madoz, Gorga, Russo. Análisis del Impacto de las TIC´s en el proceso de aprendizaje de alumnos universitarios de nivel inicial de las carreras de Informática. TICEC´05. La Plata. 2005.
[29]Madoz, Gorga. Impacto del uso de tecnología informática en el proceso de adaptación de alumnos universitarios. CACIC 2004.

Propuesta para evaluar calidad de un OA durante las etapas de su ciclo de vida

Berta E. García
Dpto de Informática – UNSL
5700- San Luis – Argentina
bgarcia@unsl.edu.ar

Hugo J. Viano
Dpto de Informática – UNSL
5700- San Luis – Argentina
hviano@unsl.edu.ar

Guillermo M. Leguizamón
Dpto de Informática – UNSL
5700- San Luis – Argentina
legui@unsl.edu.ar

Abstract

In e-learning environments, the information is an extremely valuable learning resource. Consequently, it is necessary to provide specific tools for sharing it and reuse it at a lowest possible cost. Learning objects (LO) are an effective way to re-use and exchange educational content because carry out the typical specifications and standards. This specifications and standards consider several important and closely related concepts, e.g., interoperability, accessibility, reusability, adaptability, and durability.
However, it is not sufficient that an LO technically fulfill the specifications and standards to guarantee reusability. In addition, it must provide efficient ways to access the resources (e.g., by metadata) and keep the LO quality during its cycle of life (which makes possible its use in the future).

Keywords: cycle of life, Learning Objects, LO evaluation, LO quality

Resumen

El valor de la información como recurso de aprendizaje en entornos e-learning ha generado la necesidad de disponer de ella, compartirla y reutilizarla sin que esto implique grandes costos. Desde el punto de vista técnico, las especificaciones y estándares contemplan aspectos de la reutilización, tales como: interoperabilidad, accesibilidad, reusabilidad, adaptabilidad y durabilidad. Los objetos de aprendizaje (OA) surgen con el objetivo de reutilizar e intercambiar contenido educativo. Ahora bien, que un OA cumpla técnicamente con las especificaciones y estándares, no garantiza esta reutilización. Para ello es necesario tener en cuenta, además, el acceso a los recursos, (facilitado por los metadatos), y la calidad del OA durante su ciclo de vida (lo que potencia su posibilidad de uso).

Este trabajo propone un modelo para evaluar la calidad de un OA durante las distintas etapas de su ciclo de vida.

Palabras clave: ciclo de vida, Objetos de Aprendizaje, calidad de OA, evaluación de OA

1 INTRODUCCIÓN

El concepto de Objeto de Aprendizaje (OA) constituye, dentro del e-learning, un nuevo paradigma para la creación de contenidos y actividades con propósito educativo. Si bien existen diversas concepciones acerca de qué son los OA's, resulta necesario caracterizarlos antes de establecer criterios para su evaluación.

Entre las definiciones más aceptadas en ámbitos científicos y académicos encontramos las siguientes:

> "*Objeto didáctico es cualquier recurso digital que pueda ser reutilizado como soporte para el aprendizaje*" (Wiley) [1]
>
> "*Unidad didáctica independiente y autocontenida predispuesta para su reutilización en diversos contextos educativos*" (Polsani) [2]

Entre las características sobresalientes que los OA debieran presentar para ser considerados tales, Longmire [3] tiene en cuenta las siguientes:

- Es modular, autocontenido y puede llevarse de un entorno a otro.
- No es secuencial
- Satisface un único objetivo didáctico
- Está orientado a un público amplio (puede adaptarse a destinatarios distintos a los originales)
- Es coherente y unitario dentro de un esquema predeterminado, de manera que mediante un número limitado de meta-etiquetas se pueda capturar la idea principal.
- No está en formato específico, porque puede reutilizarse sin alterar sus valores esenciales, ni su contenido.

Con la aplicación de estándares para OA se pretende que los recursos sean:

- Accesibles: habilidad de localizar y acceder componentes de aprendizaje desde una ubicación remota y distribuirlo a múltiples ubicaciones.
- Interoperables: habilidad de utilizar componentes desarrollados con un conjunto de herramientas en diferentes plataformas y/o versiones.
- Durables: habilidad de perdurar en el tiempo sin verse afectado por los cambios tecnológicos.
- Reutilizables: flexibilidad de incorporar componentes didácticos en múltiples aplicaciones y contextos.

En la actualidad el paradigma de Objetos de Aprendizaje es la aproximación teórica a la creación, distribución e intercambio de contenidos con mayor aceptación internacional dentro del panorama académico y profesional. Este modelo insiste especialmente en la posibilidad de reutilizar contenidos digitales en diferentes contextos, almacenarlos en repositorios on-line para facilitar su intercambio y fragmentarlos en unidades pequeñas que puedan unirse de maneras diferentes para formar nuevas unidades. Se han señalado, sin embargo, distintos factores que obstaculizan su intercambio por parte de los docentes [4]: la seguridad, la ausencia de filtros de calidad y la complejidad de los metadatos.

Cuando se plantea la tarea de producir OA de calidad desarrollando una propuesta de evaluación que ayude a alcanzar el máximo aprovechamiento del paradigma de los OA, tiende a suponerse que dicha propuesta debe tratar fundamentalmente aquellos aspectos relacionados con la etapa de

creación. Sin embargo, el ciclo de vida de un OA pasa por fases muy diferentes: creación, distribución, uso, rediseño y promoción.
En el apartado siguiente, se desarrollan estas etapas con más detalle.

2 EL CICLO DE VIDA DE LOS OA

Puesto que los actores que intervienen en las distintas fases del ciclo de vida de un OA son diferentes y cumplen funciones distintas, es necesario un enfoque capaz de reflejar la complejidad del proceso, que abarca cuestiones pedagógicas, organizacionales, legales y técnicas y que puede atender a las necesidades de los diversos equipos multidisciplinares que intervienen en fases sucesivas. Dicho enfoque se basa en un ciclo de vida que muestra las diferentes etapas que cada OA atraviesa [5], tal y como se muestra en la Figura 1.

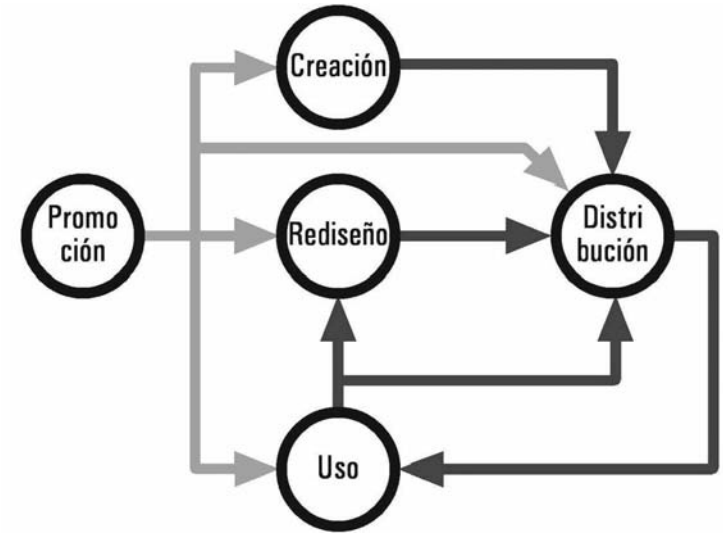


Fig. 1. *Ciclo de vida de un O A*

2.1 Creación

El ciclo comienza con la creación del OA. Existen guías de buenas prácticas [4][6] que explican los principios de una producción de contenidos basada en objetos de aprendizaje: las reglas que facilitan la reutilización, el concepto de granularidad y la necesidad de describir los materiales de acuerdo a un esquema de metadatos[1], además de otras cuestiones relevantes desde un punto de vista pedagógico y técnico: diseño gráfico, usabilidad, accesibilidad, interoperabilidad.
La creación de contenidos didácticos digitales es una tarea que requiere la participación de un equipo multidisciplinar que reúne personas con pericia técnica, miembros con experiencia en la utilización de las TIC en el ámbito educativo, y expertos con conocimientos en la materia.
La utilización de las TIC en cualquier ámbito educativo tiene evidentes implicaciones sobre el diseño instruccional, y aún más si trabajamos con OA. Uno de los problemas que surgen en este ámbito es, por ejemplo, la inquietud entre los docentes acerca de la flexibilidad de los OA. Por este motivo, la participación de personas con experiencia en uso de las TIC en el ámbito educativo es muy deseable, como así también el desarrollo de herramientas de autoría específicas que permitan esta colaboración.

2.2 Distribución

[1] Los metadatos utilizan una serie de descriptores externos normalizados o estandarizados que brindan información de los OA en formato XML, con el objetivo de conseguir interoperabilidad entre plataformas.

Los autores pueden elegir en la actualidad fórmulas de protección de su propiedad intelectual más flexibles que el copyright, basados en licencias Creative Commons [7], BC Commons [8], JISC Model [9] o similares. Portales específicos como el de JISC (Joint Information Systems Committee) ofrecen recursos interesantes para abordar esta problemática.
El objetivo de todo este abanico de modelos es encontrar un modo fiable de que los OA se distribuyan de manera fluida. No obstante, aún cuando se busquen modelos abiertos de intercambio, los derechos de propiedad intelectual son necesarios para evitar que otra persona se atribuya la autoría de OA que no le son propios.
La figura del repositorio, y el papel del gestor del repositorio, encargado de diseñarlo y mantenerlo, cobra una importancia especial en esta fase. En esta fase se deben abordar cuestiones como los criterios de evaluación de calidad de un OA, taxonomías, métodos de extracción de metadatos de OA ya creados y mecanismos de software que favorecen el intercambio en las comunidades virtuales.
Otra cuestión importante a tener en cuenta es la interoperabilidad entre plataformas.

2.3 Uso

Los OA pueden ser utilizados por estudiantes, docentes o gestores de cursos. En cuanto al uso por parte de docentes, hay algunos requerimientos básicos que resultan clave para que los OA sean empleados satisfactoriamente [5]:

- Acceso a la infraestructura tecnológica.
- Un nivel suficiente de habilidad para el uso pedagógico de las TIC.
- Tiempo de preparación suficiente.

La reusabilidad de este tipo de contenidos didácticos se ve afectada también por el contexto, el tipo de estudiantes a los que va dirigido, la modalidad de enseñanza (virtual o presencial) y su integración con diseños educativos pre-existentes. Es importante el desarrollo de instrumentos de evaluación de la calidad de los OA y su adecuación a los objetivos docentes.

2.4 Rediseño

Esta es una fase del ciclo de vida de los OA que está generando ciertos problemas. Según los resultados de CELEBRATE[10], existen pocas evidencias de que los OA se modifiquen una vez creados, con excepción de un tipo de modificación muy especial (y de gran importancia para Europa): la traducción. Cabe destacar que debido a lo novedoso de la temática, no existen OA que hayan tenido un período prolongado de uso como para que exista la necesidad de actualizar sus contenidos
Sin duda, el rediseño permite mejorar la calidad de un OA para adecuarlo a nuevos requisitos técnicos o pedagógicos reduciendo el tiempo y costo de desarrollo y favoreciendo la durabilidad e interoperabilidad del Objeto.

2.5 Promoción

Adoptar un modelo de creación e intercambio de contenidos digitales basados en OA exige cambios profundos en los procesos de funcionamiento de los centros académicos. Una de las cuestiones capitales que deben tener en cuenta es que este modelo requiere que se ofrezca apoyo y formación al profesorado. En este sentido, The Le@rning Federation [11] apuesta, en contra de la tendencia general a la formación en cascada, por modelos que enfaticen las comunidades de práctica. Estos modelos permiten organizar la formación y el apoyo entre pares de manera más efectiva.
Como en cada fase del ciclo de vida de un OA intervienen actores diferentes cumpliendo distintas funciones, y teniendo en cuenta los actores que intervienen en este proceso, la propuesta de

evaluación que se presenta analiza las fases del ciclo de vida en las que resultaría conveniente aplicar una evaluación.

3 HERRAMIENTAS PARA EVALUACIÓN DE OA

La búsqueda en un repositorio de OA basado en web puede devolver más de un objeto que permita satisfacer una determinada necesidad de aprendizaje. Para seleccionar correctamente los OA, es preciso aplicar criterios o utilizar herramientas que permitan evaluar su calidad.
Si bien existen diversas propuestas de evaluación, pueden ubicarse en extremos opuestos: la evaluación por juicio de expertos (filosofía utilizada en el repositorio MERLOT), y el uso de instrumentos de evaluación mediante cuestionarios (LORI)

MERLOT es un repositorio que almacena metadatos de los OA y los evalúa desde dos perspectivas:

1) Comentarios de los miembros de la comunidad abierta del MERLOT
2) "Peer reviews", o cometarios de expertos, que analizan el recursos en profundidad.

El inconveniente de este tipo de evaluación cualitativa es que usa el formato "texto libre", resultando poco apropiado para el tratamiento automático por sistemas de software.

Por otra parte, LORI (Learning Object Review Instrument)[12] es un cuestionario que facilita la comparación entre objetos, y provee un formato de revisión común.
Usando LORI es posible revisar OA de acuerdo a nueve aspectos. , cada uno de los cuales puede tomar valores de un rango que va de: "No aplicable" a "5".
Los ítems a evaluar son:

1- **Calidad del contenido**: veracidad, exactitud, representación balanceada de ideas, y apropiado nivel de detalle.
2- **Alineación de Objetivos de Aprendizaje:** correspondencia entre objetivos de aprendizaje, actividades, contribuciones y características de los estudiantes.
3- **Feedback y adaptación:** contenido adaptativo o manejo de feedback mediante modelado de usuarios.
4- **Motivación:** habilidad para motivar e interesar a un grupo de aprendices identificado.
5- **Diseño de la Presentación:** Diseño de la información visual y sonora para mejorar los procesos de aprendizaje.
6- **Interacción**: Facilidad de navegación, interfaz de usuario predecible y calidad de las ayudas al usuario.
7- **Accesibilidad:** diseño de controles y formatos de presentación que se adapten a estudiantes móviles.
8- **Reusabilidad:** Posibilidad de ser usado en contextos de aprendizaje variados, y con estudiantes de diferentes características.
9- **Conformidad con los estándares:** Adhesión a especificaciones y cumplimiento de estándares internacionales.

Los resultados de aplicar el cuestionario se pueden presentar en forma individual o como un promedio de los ítems evaluados. Es importante registrar los comentarios que se produzcan.

Cabe aclarar que, si bien estos modelos son opuestos, pueden complementarse para lograr mejores resultados.

4 PROPUESTA DE EVALUACIÓN

Puesto que el aprovechamiento real del modelo requiere la acción coordinada de los actores de las distintas fases del ciclo de vida de los OA, es necesario delimitar muy bien qué tipo de evaluación se realizará en cada fase.
En este caso, el modelo propuesto contempla la evaluación en cada una de las 4 fases que consideramos relevantes: Promoción, Creación, Uso y Rediseño. La figura 2 muestra las instancias de evaluación correspondientes.

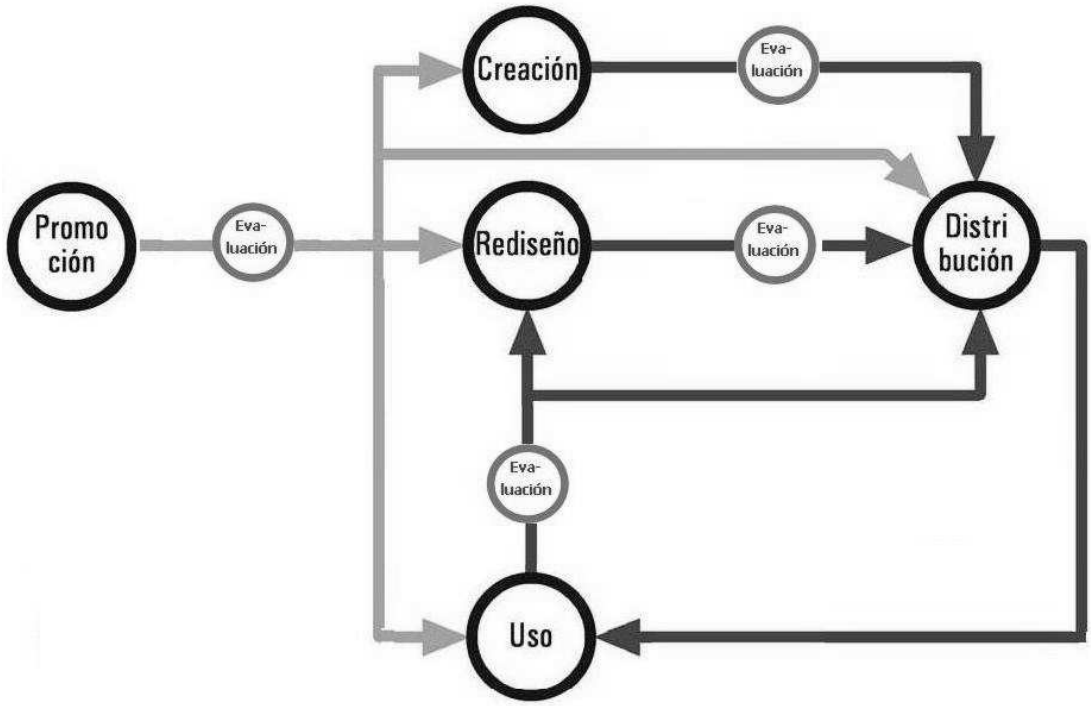


Fig. 2. *Modelo de Evaluación durante el ciclo de vida de un OA.*

Ahora bien: ¿Qué tipo de evaluación es conveniente utilizar?
Si bien es factible evaluar aplicando cualquiera de las herramientas mencionadas, proponemos lo siguiente:

- Para las fases de Promoción y Creación: utilizar el cuestionario LORI, dado que como se dijo anteriormente, analiza los recursos de forma estandarizada, y desde un punto de vista más bien técnico. Esto resulta apropiado para estas fases, puesto que es muy probable que el OA no haya sido utilizado aún y la evaluación la realice el mismo autor.
- Para la fase de Uso, es conveniente la evaluación del OA por parte de usuarios distintos al autor. Por ello recomendamos Peer Review, lo que puede derivar en el Rediseño del mismo.
- Sin embargo, para la fase de Rediseño proponemos un modelo mixto, aplicando LORI y Peer Review, ya que el OA ha sido utilizado al menos una vez.

Este modelo teórico será usado para evaluar los OA que forman parte del Repositorio de Objetos Informáticos (ROI). Este gestor permite, desde el apartado "Comentarios", reflejar la información obtenida por los Peer Reviews. Está en etapa de estudio la incorporación de los resultados del cuestionario LORI mediante el llenado de los metadatos del OA. Utilizando el esquema LOM [13], es posible vincular mediante anotaciones, (categoría de metadatos 8 Annotation) diferentes instancias evaluativas, e identificar la fase a la que corresponden.

5 CONCLUSIONES Y TRABAJOS FUTUROS

Si bien la evaluación es una opción más que interesante para aplicar a los OA durante su ciclo de vida cabe destacar que, debido a lo novedoso de la temática, éstos se encuentran en su mayoría en etapa de Promoción o Creación.

En nuestro caso particular, el grupo de trabajo se encuentra en la etapa de desarrollo e implementación de un repositorio de OA informáticos (ROI), cuya finalidad es el intercambio entre materias de carreras de grado en informática. Si bien hay OA desarrollados, aún no se ha comenzado con la etapa de distribución hasta tanto se tenga el repositorio ROI a punto. Debido a esto, las herramientas de evaluación aplicadas hasta el momento han sido cuestionarios LORI. No obstante, se prevé la aplicación del modelo de evaluación de OA durante su ciclo de vida, integrando el repositorio ROI en la etapa de distribución.

Consideramos que estas prácticas redundarán en beneficios en cuanto a la calidad de los recursos educativos a compartir entre diferentes docentes de Informática.

6 BIBLIOGRAFÍA

[1] WILEY, D. (2002). "Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy", AIT/AECT, The Instructional Use of Learning Objects, Association for Instructional Technology, 1-35

[2] Polsani, P. R. (2003). Use and Abuse of Reusable Learning Objects. Journal of Digital Information, volumen 3, número 4, artículo No. 164. [Consultado el 7/7/2007] Disponible en: http://jodi.tamu.edu/Articles/v03/i04/Polsani/

[3] Longmire, W. (2000). A primer on learning objects. ASTD Learning Circuits. [Consultado el 7/7/2007] Disponible en: http://www.learningcircuits.org/2000/mar2000/Longmire.htm

[4] Becta (2005): Packaging and publishing Learning Objects: Best practices guidelines. [Consultado el 22/6/2007Disponible en:

http://www.becta.org.uk/page_documents/industry/content_packaging.pdf

[5] Azpeitia, I.; Monge, S.; Ovelar, R. (2005).: "Una Aproximación al Diseño de una Guía de Buenas Prácticas en torno al Paradigma de los Learning Objects." Comunicado presentado en el II Simposio Pluridisciplinar de Diseño, Evaluación y Descripción de Contenidos Educativos Reutilizables [Consultado el 7/7/2007] http://www.uoc.edu/symposia/spdece05/pdf/ID14.pdf

[6] Smith, R. S (2004): Guidelines for authors of Learning Objects. The New Media Consortium. Austin [Documento online consultado el 22/6/2007].Disponible en: www.nmc.org/guidelines/

[7]. Creative Commons. [Consultado 20/6/2005] Disponible en: http://creativecommons.org/

[8] BC Commons License (v 1.1). [Consultado 29/6/2005] Disponible en: http://www.bccampus.ca/AssetFactory.aspx?did=25

[9] Jisc Model License. [Consultado 11/7/2007] http://www.jisc.ac.uk/coll_guide_jiscmodel.html

[10] McCormick, R., Scrimshaw, P., Li, N., Clifford, C. (2004): CELEBRATE Evaluation report (version 2). [Consultado el 30/6/2007] Disponible en: http://www.eun.org/eun.org2/eun/Include_to_content/celebrate/file/Deliverable7_2E valuationReport02Dec04.pdf

[11] Muirhead, H.: An Assessment of the Learning Objects, Models and Frameworks. The Le@rning Federation Schools Online Curriculum Content Initiative Australia. (2004). [Consultado el 30/6/2007] Disponible en:

http://www.thelearningfederation.edu.au/tlf2/sitefiles/assets/docs/muirhead_haughey_0105.pdf

[12] Nesbit, J; Belfer, K. ; Leacock T. LORI-Learning Object Review Instrument. User Manual Versión1.5. [Consultado el 7/7/2007] Disponible en: http://www.elera.net/eLera/Home/Articles/LORI%201.5.pdf

[13] IEEE Draft Standard for Learning Object Metadata. [Consultado el 20/7/2007] Disponible en: http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

ROI: Repositorio de Objetos de Aprendizaje Informáticos

Viviana Mercedes Ponce, Irma Guadalupe Pianucci, Marcela Cristina Chiarani
Departamento de Informática, Universidad Nacional de San Luis
(5700) San Luis, Argentina

vmponce@unsl.edu.ar, pianucci@unsl.edu.ar, mcchi@unsl.edu.ar

Proyecto: "Herramientas Informáticas Avanzadas para Gestión de Contenido de Carreras de Grado en Informática"

Abstract

This work presents the advance in the development of a learning objects repository for computer science (ROI) that facilitates sharing materials and experiences. On the present time, one of the approaches more widely accepted in the application of the information technologies to education is based in a new paradigm of designs of learning activities, putting emphasis in the reusability of contents and activities in different surroundings. As consequence of this arises the necessity to store, to look for, to recover, to consult and to download these learning objects.

Considering the common problems in the admission to computer science career at different universities in our country, and facing the effort that implies the development of learning materials, this group have the aim to investigate and develop software tools that allow the creation of a knowledge network for Computer Science careers in order to optimize the resources generation and their reusability.

Resumen

En este artículo se presenta el avance en el desarrollo de un repositorio de objetos de aprendizaje de informática (ROI) que facilite compartir materiales y experiencias.

En la actualidad, uno de los enfoques más ampliamente aceptados en la aplicación de las tecnologías de la información a la educación se basa en un nuevo paradigma de diseño de actividades de aprendizaje, poniendo énfasis en la reutilización de contenidos y actividades en distintos entornos. Como consecuencia surge la necesidad de almacenar, buscar, recuperar, consultar y bajar estos objetos de aprendizaje.

Teniendo en cuenta la problemática en común que se presenta en el ingreso a carreras de Informática de distintas universidades de nuestro país, y considerando el esfuerzo que implica que los docentes realicen materiales educativos pertinentes a tal fin, éste grupo tiene como objetivo investigar y desarrollar herramientas de software que permitan crear una red de conocimientos localizados en carreras de grado en informática. Se busca con ello propiciar un marco de colaboración para el desarrollo de las actividades académicas en carreras afines a modo de optimizar la generación y reusabilidad de recursos.

Palabras claves: Objetos de Aprendizaje, Repositorios de Objetos de Aprendizaje, LMS, Diseño de Aprendizaje.

1. Introducción

Como explica Sicilia [9], al principio de la década de los 90 se popularizo la divulgación de contenidos educativos a través de Internet. Surge a su vez la creación de los repositorios de objetos de aprendizaje, entendidos como auténticos centros de intercambio de materiales didácticos. Previsiblemente éstos, se convertirán en uno de los pilares del elearning del futuro más inmediato. Básicamente se pueden entender como: almacenes digitales en los que se recogen aportaciones individuales de los miembros de una comunidad para ser compartidos y evaluados entre todos ellos.

Haciendo propia la problemática en común que se presenta en el ingreso a carreras de Informática de distintas universidades de nuestro país[1], y considerando el esfuerzo que implica que los docentes realicen materiales educativos pertinentes a tal fin, éste grupo tiene como objetivo investigar y desarrollar herramientas de software que permitan crear una red de conocimientos localizados en carreras de grado en informática. Se busca con ello propiciar un marco de colaboración para el desarrollo de las actividades académicas en carreras afines a modo de optimizar la generación y reusabilidad de recursos.

El presente trabajo está organizado de la siguiente manera: comienza con una introducción al tema, seguida de la conceptualización de software libre. Continúa con repositorios de código libre. Luego, se detalla el modelo de repositorio que desarrollo nuestro proyecto. Finalizamos este documento con la conclusión del mismo

2. ¿Porque elegir software libre?

Nuestro proyecto de investigación en un todo de acuerdo con las premisas del software libre como lo expresa en su libro digital Richard M. Stallman:

> "*Creemos que los usuarios de ordenadores deberían tener libertad para cambiar y redistribuir el software que utilizan. El adjetivo «libre» en el software libre hace referencia a la libertad: libertad del usuario para ejecutar, modificar y redistribuir software. El software libre contribuye al saber humano, al contrario que el software propietario. Por este motivo, las* ***universidades*** *deberían fomentar el software libre, para hacer una aportación al progreso del conocimiento humano, del mismo modo que deben animar a científicos y académicos a publicar sus obras."* [2]

El mismo autor aporta una definición difundida ampliamente en Internet, que aclara cual es la libertad que tienen los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Allí también, menciona las cuatro libertades para los usuarios del software:

De acuerdo con tal definición, el software es "libre" si garantiza las siguientes libertades:

- **libertad 0**: ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.)
- **libertad 1**: estudiar y modificar el programa (para lo cuál es necesario poder acceder al código fuente)
- **libertad 2**: copiar el programa de manera que se pueda ayudar al vecino o a cualquiera
- **libertad 3**: mejorar el programa, y hacer públicas las mejoras, de forma que se beneficie toda la comunidad.

Es importante señalar que las libertades 1 y 3 obligan a que se tenga acceso al código fuente. La libertad 2 hace referencia a la libertad de modificar y redistribuir el software libremente licenciado bajo algún tipo de licencia de software libre que beneficie a la comunidad.

De lo antes dicho se desprende que para las instituciones educativas públicas es indiscutible la conveniencia de trabajar con material de estas características.

3. Repositorios de código libre

El número de comunidades virtuales que han surgido relativas al desarrollo de repositorios y Objetos de Aprendizaje demuestran un gran interés tanto en el ámbito académico como en el comercial.

Los siguientes proyectos se encuentran trabajando con la perspectiva del Software libre: Aloha II[23], Pool[24], Eduplone[25], eRib[11], Planet[21], DOOR[10]**.** Entre los repositorios desarrollados de primera generación se encontró Careo[12], uno de los más conocidos que estuvo disponible de este modo.

De la evaluación realizada en relación a Repositorios se procuró seleccionar uno que se adecue más a nuestras necesidades, presentadas en [6], por tal motivo, se optó por DOOR. El motivo fundamental de esta selección es que la plataforma por la que fue desarrollado coincide con la nuestra. Actualmente en el proyecto se cuenta con un servidor Linux con PHP y MySql, en el que se alojará el repositorio finalmente desarrollado por nuestro grupo.

DOOR se desarrolló siguiendo el estándar internacional y sus metadatos, IMS Metadata 1.2.1 y Content Package 1.1.3.

4. Modificaciones al Repositorio

Dado que el repositorio se encontraba disponible en los idiomas italianos, ingles y francés se realizó la traducción al español.

El usuario accede a los servicios del repositorio, siendo éste el que se encarga de realizar las operaciones con los objetos de aprendizaje.

Los servicios varían según el rol de usuario. Los que se detallan a continuación son los servicios disponibles:

1. *Agregar objetos al repositorio*.
   Los objetos de aprendizaje se pueden agregar en forma de paquetes que incluyen la descripción del objeto y el contenido o bien incluir solamente los metadatos junto con la url del objeto.
2. *Buscar objetos en el repositorio*.
   La búsqueda se realiza en base al texto libre o bien a partir de distintos parámetros preestablecidos.
3. *Extraer objetos del repositorio*.
   Esta opción entrega el paquete completo que incluye tanto a los metadatos como al contenido mismo.

En la pantalla principal del repositorio ROI se puede observar que el acceso al mismo se realiza indicando usuario y contraseña. Si el usuario no se ha registrado puede generar una cuenta para acceder. La función **Agregar nuevo Usuario** se incorporó dado que el software original no lo contemplaba. Si bien esta opción se realiza por default con el rol de invitado, el administrador es el que puede cambiar el tipo de usuario. Los tipos de usuarios que se pueden generar son los siguientes: invitado, autor y administrador.

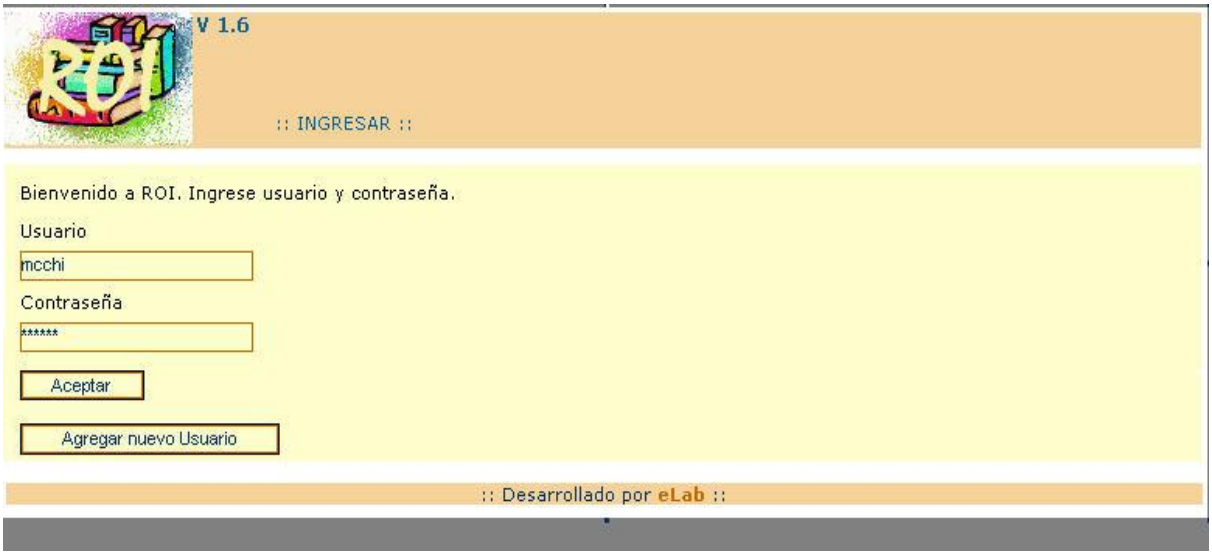


Figura 1

4.1. La interfaz del usuario autor

A continuación se visualiza en la figura 2 la interfaz correspondiente al rol de usuario autor. Tiene un menú en la parte superior con cuatro opciones, estas son: Principal, búsqueda, explorar y salir.

La **opción principal** permite agregar e importar objetos y actualizar el perfil del usuario. Claramente orienta al usuario en las posibilidades disponibles en el repositorio a través de una ayuda rápida en la pantalla

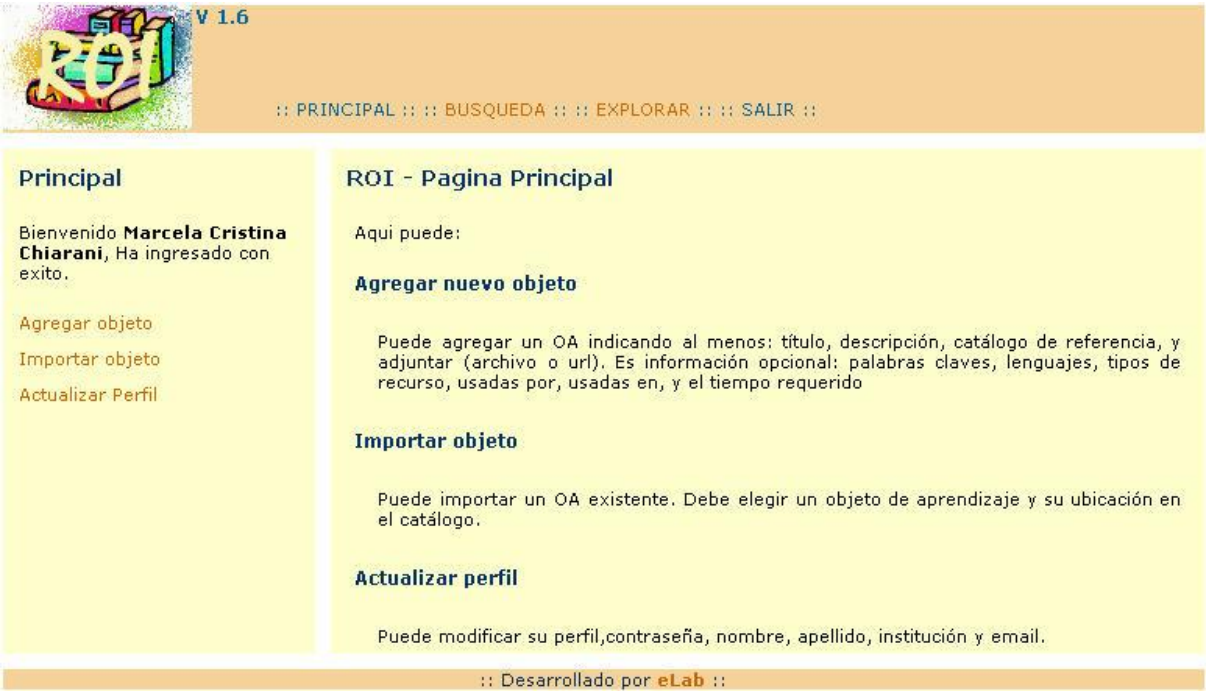


Figura 2

La segunda opción disponible para el autor es la **Búsqueda de objetos de aprendizaje** en el repositorio. En la figura 3 se observa que la misma puede ser a través de un texto libre, o a partir de

palabras claves prefijadas en los metadatos. Para realizar esta tarea existe una opción para establecer las mismas.

Figura 3

Asimismo, se permite una vez localizado el objeto corregirlo o suprimirlo. Teniendo en cuenta que esto se puede realizar en los objetos en los que él es propietario.

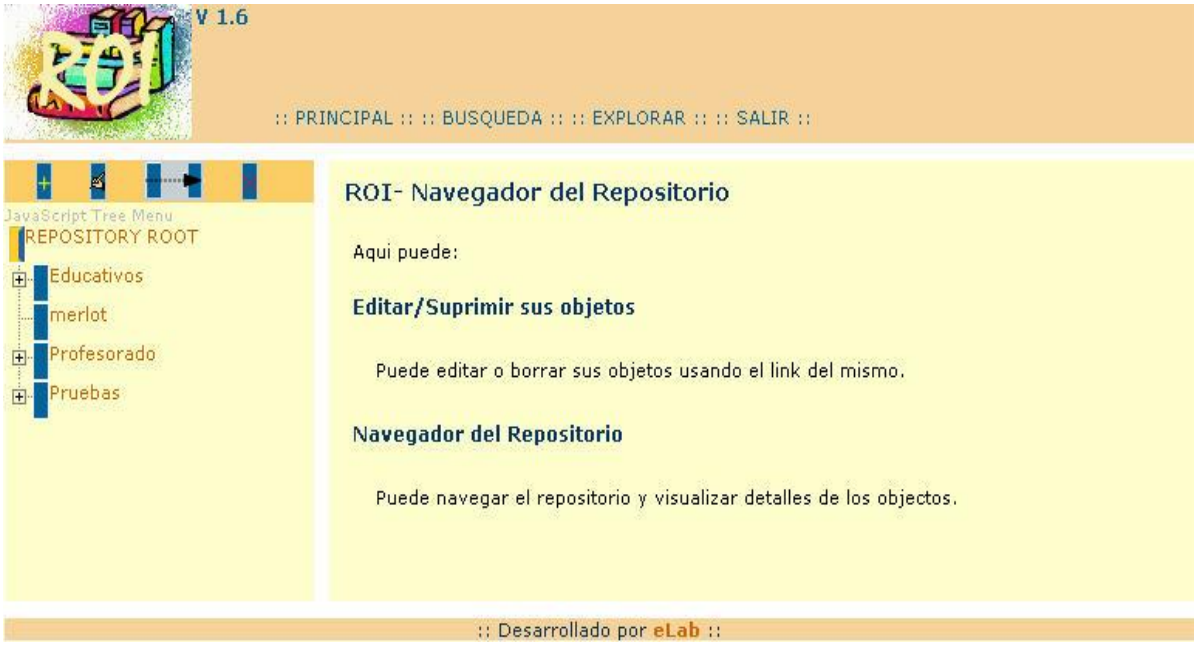


Figura 4

La tercera opción, **Explorar**, permite navegar por todo el mapa del repositorio. Pudiendo a partir del link de los objetos suprimirlos o Editar su contenido. Se muestra en la figura 4.

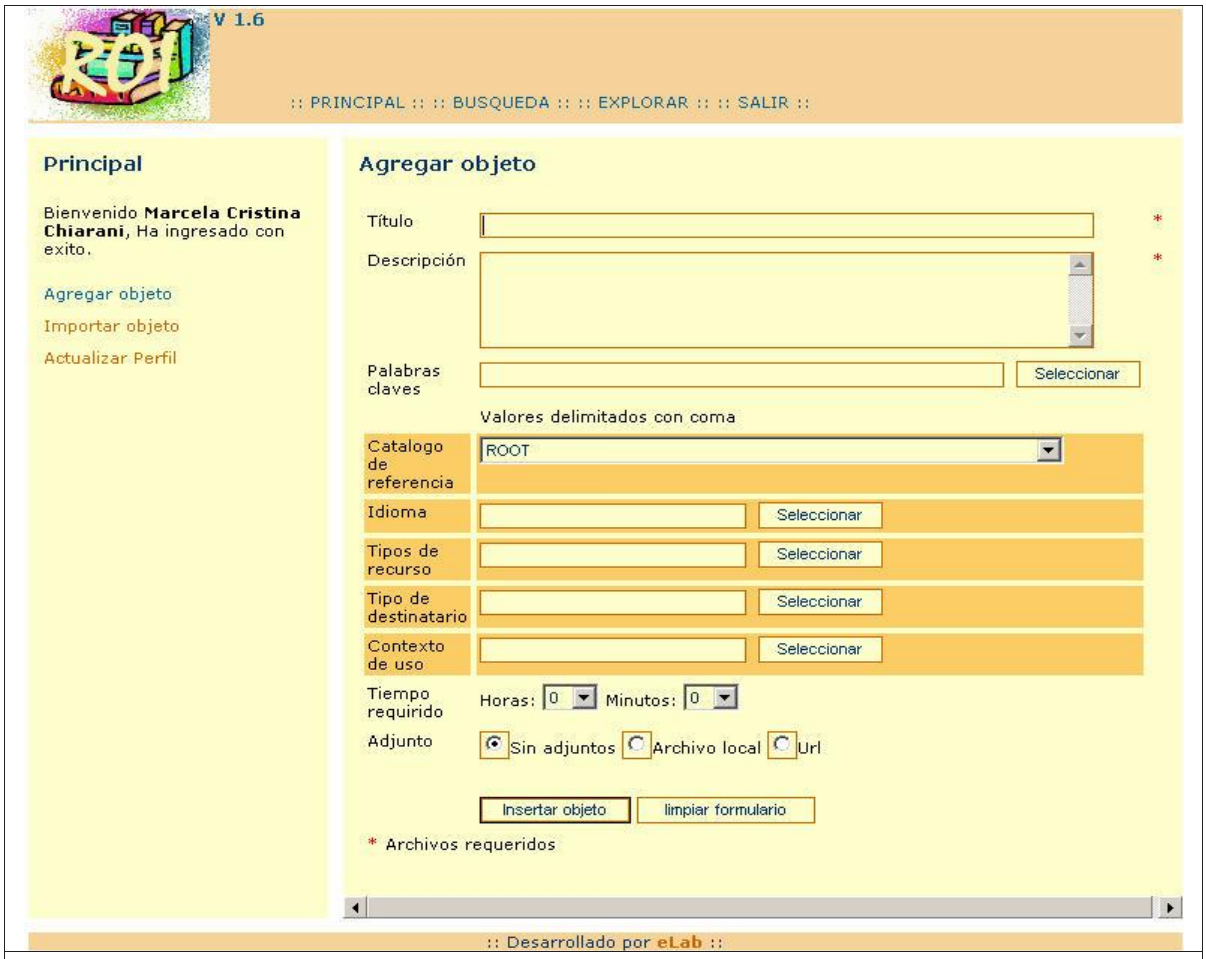


Figura 5

Sin duda la función más significativa del repositorio es la incorporación de objetos al mismo. En la figura 5 se observa que los datos obligatorios a llenar son el título y la descripción, quedando como opcionales los demás campos. Es importante destacar que el objeto no necesariamente se incorpora en esta etapa.

5. Conclusiones y tareas futuras

Actualmente nuestro grupo de trabajo ha concluido la fase de instalación y testing de ROI. Se espera continuar con las modificaciones para que se adecue a nuestros propósitos.

Además, se está trabajando en la creación de Objetos y Unidades de Aprendizaje que servirán para analizar el comportamiento del repositorio.

Al finalizar el proyecto, se espera poner online esta herramienta y permitir su acceso desde la plataforma Ilias, nucleando la producción académica de los docentes.

Nuestro equipo de trabajo busca compartir experiencias y conocimientos relacionados con nuestro tema central de investigación para posibilitar espacios de discusión cooperativos.

6. Bibliografía y referencias

[1] Cataldi Z. Figueroa N, Salgueiro F, Rendón J., Costa G., Lage F., Perichinsky G (2004). Los estilos de aprendizaje y el desgranamiento Universitario en Ingeniería Informática. CACIC 2004. La Matanza.
[2] Richard M. Stallman. El Movimiento Del Software Libre.
[3] Learning Resource Meta-Data Specification: Versión 1.2.4 IMS Global Leraning Consortium. http://www.imsproject.org/metadata/index.html
[4] Alvarez González L., Gallardo González M (2004). Diseño de un Repositorio de Objetos de Apoyo al Aprendizaje Colaborativo. CISCI 2004. Orlando. USA.
http://www.inf.uach.cl/lalvarez/publicaciones/CISCI_2004.pdf
[5] Friesen, N. (2005) Interoperability and Learning Objects: an overview of elearning standardization. Int. Journal of Knowledge and Learning Objects, 1.
[6] Chiarani M., Leguizamon G., Pianucci I. (2006). Repositorio de Objetos de Aprendizaje para Carreras Informáticas .WICC 2006, Moron.
[7] Leguizamon G., Lucero Ma., Viano H., García B. (2005). Generación de contenidos de aprendizaje estandarizados para plataformas de e-learning. CACIC 2005 – Concordia.
[8] Software libre. Free software foundation. http://www.fsf.org/
[9] Sicilia Urbán Miguel-Angel, Sánchez Alonso Salvador. (2006) Repositorios de objetos de aprendizaje. Information Engineering Research Unit. Universidad de Alcalá.
[10] DOOR. http://door.elearninglab.org/website/index_ita.php
[11] eRIB. Repositorio Edusource. http://edusource.licef.teluq.uquebec.ca/ese/fr/install_erib.htm
[12] Careo. Repositorio del campus de Alberta. http://www.careo.org/documents/overview.html
[13] Apollo. http://apollo.ucalgary.ca
[14] IEEE Learning Technology Standards Committee (2002) Learning Object Metadata (LOM), IEEE 1484.12.1-2002.
[15] Iriarte Navarro, L. et al. - Generación de una biblioteca de objetos de aprendizaje (LO) a partir de contenidos preexistentes. Publicado en revista electrónica Red (Revista de Educación a Distancia) - Febrero de 2005. http://www.um.es/ead/red/M2/
[16] DLearn. https://www.dlearn.arizona.edu/
[17] García B., Leguizamón G., Viano H. (2006). Utilización de una Unidad de aprendizaje sobre Mapas Conceptuales en contenidos curriculares de materias de los Profesorados en Computación y en Tecnología. WICC 2006, Moron.
[18] IMS Learning Design. Information Model, Best Practice and Implementation Guide, XML Binding. www.imsglobal.org/content/learningdesign
[19] Ariadne. www.ariadne-eu.org/
[20] Sánchez Arias, V. Diseño del patrimonio de recursos educativos basados en una red de acervos abiertos y distribuidos de objetos de aprendizaje. Taller sobre tecnología de Objetos de Aprendizaje (TOA) - 4o. Encuentro Internacional de Ciencias de la Computación (ENC' 03)
http://www.comunidades.ipn.mx/Portal/Languages/Espa%C3%B1ol/UploadFiles/Documents/52victor%20lania.pdf
[21] Planet. http://ants.etse.urv.es/planetdr
[22] Repositorio institucional E-Print. http://www.eprints.org/
[23] Aloha II. http://aloha.netera.ca
[24] Pool. Portal para Edusplash. http://edusplash.net
[25] Eduplone. http://www.eduplone.net
[26] Repositorio digital DSpace. http://www.dspace.org/

Sistema Experto aplicado a la Enseñanza de Ciclos Termodinámicos

Vanina E. Beraudo
Facultad de Ingeniería - UNLPam
vaninaberaudo@gmail.com

Luis A. Gago
Facultad de Ingeniería - UNLPam
gagoluis@ing.unlpam.edu.ar

Martín H. Echeverria
Facultad de Ingeniería - UNLPam
echeverriamh@gmail.com

Natalia S. Stark
Facultad de Ingeniería - UNLPam
nstark@ing.unlpam.edu.ar

Abstract

This article deals with an *expert system* application capable of determining the feasibility of operation of a thermic machine from the thermodynamic point of view, by means of the analysis of the design of a power cycle, the *Joule Brayton* cycle. Also, it intends to give Thermodynamic students an innovative tool which allows them to learn the fundamentals on the basis of their successes and mistakes, by spotting physically impossible designs, offering them the explanations that a good laboratory assistant would provide.

Keywoards: Expert Systems, Thermodynamic, Thermodynamic Cycles, Knowledge Engineering

Resumen

El presente artículo trata la aplicación de un *sistema experto* capaz de determinar la factibilidad de funcionamiento de una máquina térmica desde el punto de vista termodinámico, mediante el análisis del diseño de un ciclo de potencia, el *ciclo Joule Brayton*; y ofrecer a estudiantes de Termodinámica, en especial, una herramienta innovadora que les permita aprender principios fundamentales del dominio, en base tanto a sus aciertos como a sus errores, detectando diseños físicamente imposibles y proporcionándoles las explicaciones que un buen asistente de laboratorio les brindaría.
Se describen fases y etapas de la metodología empleada para construir el sistema, a fin de brindar al lector una idea general del mismo y de cómo el sistema trabaja.

Palabras Claves: Sistemas Expertos, Termodinámica, Ciclos Termodinámicos, Ingeniería del Conocimiento

1 INTRODUCCIÓN

Pedagógicamente, la enseñanza y el aprendizaje del análisis y diseño de ciclos termodinámicos es un problema importante [2], razonarlos requiere un amplio y profundo entendimiento de los fundamentos de Termodinámica, por lo cual el estudio de ciclos ocupa la mayor parte del entrenamiento práctico de estudiantes de ingeniería en las cátedras de Termodinámica. La experiencia en diseño se considera esencial para la educación en ingeniería y provee un contexto motivador poderoso para el aprendizaje de los principios físicos fundamentales: no se puede diseñar un motor de reacción, un refrigerador, o planta de potencia sin usar un amplio rango de principios físicos. Esta experiencia es difícil de transmitir en un aula típica porque muchos artefactos físicos interesantes (como plantas de poder, artefactos del motor de reacción, y refrigeradores) son caros o peligrosos para construir y experimentar con ellos.
Los Sistemas Basados en Conocimiento ó Sistemas Expertos (SE) emulan el comportamiento humano experto en un área de conocimiento determinada. Constituyen sistemas de ayuda a la toma de decisiones en áreas tan diversas como la selección de estrategias instruccionales [5], el control de variables ambientales [6], la configuración de ventiladores en neonatolgía [7], entre otras.
Este trabajo presenta una propuesta innovadora para el estudio de los ciclos termodinámicos, un SE que permite determinar la factibilidad de funcionamiento de una máquina térmica, mediante el análisis y diseño del ciclo Joule Brayton.
El ciclo Brayton es un ciclo de potencia de gas y es la base de las turbinas de gas. Tiene como función transformar energía que recibe en forma de calor en trabajo mecánico o potencia, si se lo refiere a unidades de tiempo.
El Sistema Experto sirve de guía y en especial de herramienta al usuario que comienza a trabajar con ciclos termodinámicos, detectando diseños imposibles de construir, en menos tiempo, especificando la causas que imposibilitan la construcción y ofreciendo la posibilidad de realizar cambios y /o ajustes sobre los valores para optimizar su rendimiento tal como lo haría una persona idónea en el tema.
El trabajo se organiza como sigue: la sección 2 describe el problema pedagógico que motiva la construcción del SE e incluye una breve descripción de qué son los ciclos termodinámicos y cómo ellos trabajan. La sección 3 presenta la descripción general de la solución propuesta al problema planteado en la sección anterior. La sección 4 detalla el proceso de adquisición de conocimientos. La etapa de conceptualización realizada a fin de obtener una representación de los conocimientos del experto se describe en la sección 5. La formalización de los conocimientos se expone en la sección 6. Posteriormente, la sección 7, muestra la etapa de implementación del sistema y la evaluación realizada al mismo. Finalmente, la sección 8 presenta las conclusiones y trabajos futuros.

2 LA TAREA: ENSEÑAR EL ANÁLISIS Y DISEÑO DE CICLOS TERMODINÁMICOS

En Termodinámica hay dos importantes áreas de aplicación, la generación de potencia y la refrigeración [1]. Ambas se realizan mediante sistemas que operan en un ciclo termodinámico, dentro del cual un fluido de trabajo sufre una serie de transformaciones para procesar energía.
Los ciclos se dividen en dos categorías generales: ciclos de potencia y ciclos de refrigeración. Los dispositivos o sistemas empleados en producir una salida de potencia neta reciben el nombre de máquinas térmicas, y los ciclos termodinámicos que operan se denominan ciclos de potencia. Los dispositivos o sistemas utilizados en producir refrigeración se llaman refrigeradores, acondicionadores de aire o bombas de calor, y los ciclos termodinámicos que operan reciben el nombre de ciclos de refrigeración.

Los ciclos termodinámicos también se categorizan como ciclos de gas o ciclos de vapor, dependiendo de la fase del fluido de trabajo, es decir, del estado de la sustancia que circula por el sistema.
Los ciclos termodinámicos pueden, incluso, categorizarse como ciclos cerrados o abiertos. En los ciclos cerrados el fluido de trabajo es regresado a su estado inicial al final de cada ciclo y se recircula. En los ciclos abiertos el fluido de trabajo se renueva al final de cada ciclo, en lugar de ser recirculado.
Para los ingenieros en Termodinámica, los ciclos termodinámicos juegan el mismo rol que los circuitos electrónicos para los ingenieros en electrónica: una cierta cantidad de partes, compresores, turbinas, intercambiadores de calor, son combinadas en red generando alternativas de diseño para un problema dado.
La Figura 1 muestra el diseño de un ciclo Joule Brayton cerrado. El ciclo opera de la siguiente manera: el fluido de trabajo entra a la cámara de combustión con la temperatura elevada en el estado *e2*, donde se le agrega energía mediante un proceso de presión constante, hasta que alcanza la temperatura elevada del estado *e3*. Entonces, el fluido entra a la turbina y tiene lugar una expansión isentrópica, produciendo cierta potencia. El fluido sale de la turbina al estado *e4* y pasa a ser enfriado, en un proceso a presión constante. En el enfriador, la temperatura baja, de donde sale al estado *e1*, listo para entrar al compresor. Ahí el fluido es comprimido isentrópicamente al estado *e2* y el ciclo se repite.

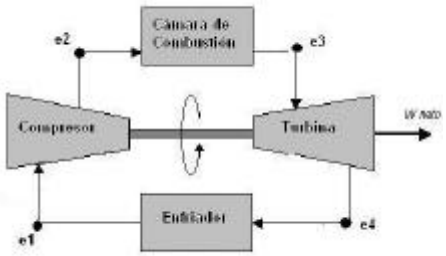


Figura 1: Ciclo Joule Brayton

El análisis de ciclos responde a cuestiones tales como la eficiencia global del sistema, cuánto calor o trabajo es consumido y/o producido, que propiedades del fluido (temperatura, presión, volumen) son requeridos por los componentes o dispositivos, así como entender de qué manera las propiedades de los componentes y del fluido en distintos puntos del ciclo afectan a las propiedades globales del mismo. Distintos diagramas constituyen auxiliares valiosos en el análisis de procesos termodinámicos. En especial, el diagrama de propiedades Temperatura-entropía (T-s) y el diagrama de Exergías.
En Termodinámica, la mayor parte de los dispositivos que producen potencia operan en ciclos. Los ciclos que se efectúan en los dispositivos reales son difíciles de analizar por la presencia de efectos complicados, como la fricción, y la ausencia de tiempo suficiente para establecer las condiciones de equilibrio durante el ciclo. Para hacer factible el estudio analítico de un ciclo, es necesario conservar las complejidades en un nivel manejable y utilizar algunas idealizaciones (modelado). Cuando al ciclo se le eliminan todas las irreversibilidades y complejidades internas, se finaliza con un ciclo que se asemeja al ciclo real pero conformado por completo por procesos internamente reversibles. Un ciclo de estas características recibe el nombre de ciclo ideal. Un modelo idealizado simplemente permite a los ingenieros estudiar los efectos de los principales parámetros que gobiernan el ciclo, sin detenerse en los detalles. No siempre las conclusiones del análisis de ciclos ideales son aplicables a los ciclos reales.

Termodinámica es una materia importante en la formación de un futuro ingeniero. En su currícula, los ciclos termodinámicos constituyen un tema interesante, dado su carácter integrador de principios y por sus posibilidades de aplicación en la práctica. Su entendimiento requiere un amplio y profundo conocimiento de los principios físicos que fundamentan la materia. De hecho los libros más introductorios a la Termodinámica dedican diferentes capítulos al análisis de ciclos y hasta algunos libros se dedican únicamente al análisis de ciclos.
Una variedad de problemas aparecen cuando se enseña a los estudiantes como diseñar y analizar ciclos [4]: (1) los estudiantes tienden a retrasarse en el mecanismo de resolver ecuaciones y cálculos rutinarios. Esto impide explorar múltiples alternativas de diseño y otros estudios, por ejemplo: ver como la eficiencia varía como función de la eficiencia de la turbina versus como ésta varía como una función de la temperatura de salida de la caldera. De modo que sin hacer este estudio comparativo algunas oportunidades de aprendizaje lamentablemente se pierden. (2) los estudiantes están preocupados respecto a qué suposiciones para el modelado necesitan hacer, tal como asumir que un intercambiador opera isobáricamente, o que una válvula lo hace isoentálpicamente, (3) los estudiantes frecuentemente no cambian los parámetros que eligieron para ver si sus diseños son físicamente posibles, por ejemplo: que sus diseños no requieran el absurdo de una bomba que produzca en lugar de consumir trabajo. (4) Si un diseño no es factible, dado el costo de tiempo y esfuerzo que requiere cambiar parámetros y recalcular valores, éstos no se modifican para hacer el diseño factible.

3 PROPUESTA

Para contribuir a enriquecer el proceso de enseñanza-aprendizaje, se propone un Sistema Experto (SE) construido específicamente para ayudar a los estudiantes a aprender ingeniería termodinámica. Proveyendo un ambiente de aprendizaje que maneje cálculos rutinarios, facilidades de análisis, ayude a los estudiantes a mantener pistas del modelo de suposición y detecte diseños físicamente imposibles, omisión de datos, inconsistencias o incompatibilidades especificando las causas y/o sugiriendo mejoras correctivas.
El SE diagnosticará la factibilidad de funcionamiento de un ciclo de potencia, el ciclo Joule Brayton
Las tareas propias de un experto que realiza el sistema son, principalmente:

- *Diseño*: en el sentido de que el usuario podrá variar un conjunto de posibles alternativas (las propiedades del fluido, de los dispositivos, de los estados y del ciclo) y el sistema en base a las relaciones de los distintos parámetros involucrados en el diseño de la máquina verificará que estén de acuerdo con determinadas restricciones. Alertará al usuario sobre inconsistencias o contradicciones en datos aportados justificando el porqué de las mismas.
- *Diagnóstico*: en el sentido de que una vez analizado el diseño, el sistema determinará las posibilidades de funcionamiento de la máquina térmica. (encontrar fallas futuras)
- *Depuración, reparación:* en el sentido de recomendar acciones correctivas.
- *Ayudante inteligente*: en el sentido de aconsejar, proporcionar información o efectuar tareas como resolver cálculos rutinarios y realizar distintos análisis termodinámicos.

Antes de comenzar a desarrollar el sistema, se efectúo un análisis de viabilidad a fin de asegurar la conveniencia de resolver el problema a través del empleo de un SE, con resultados satisfactorios.
El desarrollo se llevó adelante empleando fases y etapas de la metodología del área de Ingeniería en Conocimiento denominada I.D.E.A.L [3] (acrónimo de las fases que la conforman: Identificación de la tarea, Desarrollo del prototipo, Ejecución de la construcción del sistema integrado, Actuación para conseguir el mantenimiento perfectivo, Lograr una adecuada transferencia tecnológico).

4 ADQUISICIÓN DE CONOCIMIENTOS

El proceso de adquisición de conocimientos (AC) comienza con la recolección de información y continúa en paralelo a todas las etapas de construcción del sistema. Esta adquisición en sus dos facetas, extracción de los conocimientos públicos de sus fuentes y educción de los conocimientos privados del experto, se alterna cíclicamente con la etapa de conceptualización (explicada en la próxima sección) para modelizar el comportamiento del experto.
Para tratar de superar el problema de extraer los conocimientos correctos y obtener la cantidad y el tipo adecuado de éstos, se llevó adelante una planificación y control del proceso de AC.

4.1. Extracción de Conocimientos

Cuando la fuente de conocimientos se encuentra en forma escrita, el proceso de AC se denomina extracción de conocimientos. La extracción se realizó tanto de fuentes públicas (libros sobre Termodinámica, Internet, principios de la Termodinámica) como de fuentes privadas (apuntes propios del experto, tablas y gráficos, lectura y análisis de ejercicios resueltos por el experto, utilizados en la cátedra de Termodinámica, lectura de diversas publicaciones sobre técnicas de Inteligencia Artificial aplicadas a dominios de física, en especial física termodinámica, etc.).

4.2 Educción de Conocimientos

Cuando los conocimientos se obtienen de los seres humanos, el proceso de AC se denomina educción de conocimientos. Se comenzó esta tarea con un interrogatorio inicial para obtener una visión de alto nivel del dominio y para comprender su alcance, la tarea que realiza el experto y el entorno de la tarea. Posteriormente se llevó a cabo una investigación profunda de forma gradual para obtener los detalles concretos de las distintas áreas del dominio.
Para realizar el trabajo se emplearon distintas técnicas de educción, tanto directas como indirectas:

- *Entrevista Abierta*: donde se le planteó al experto algunas preguntas generales acerca de algún tema en particular de su trabajo. Esta técnica resultó especialmente útil al principio de la etapa de educción para entender la tarea que realiza el experto así como también cada vez que se abordaba algún nuevo punto en su trabajo para lograr una comprensión general del mismo.
- *Entrevista Estructurada*: Esta técnica fue muy útil para ampliar ó profundizar los conocimientos adquiridos con las entrevistas abiertas. Se le planteó al experto una serie de preguntas cerradas y con esto se pudo adquirir conocimientos más específicos y recabar información faltante.
- *Observación de Tareas Habituales*: Se le pidió al experto que resuelva problemas de termodinámica y se observó como lo realizaba. Esta técnica permitió ver que pasos seguía el experto para realizar su tarea y poder distinguir que datos eran importantes y cuales secundarios.
- *Incidentes Críticos*: Se le pidió al experto que comente casos especialmente interesantes o difíciles que se le hayan presentado y que describa cómo los resolvió.
- *Clasificación de Conceptos*: Con los conceptos adquiridos en sesiones anteriores de AC se le pidió al experto que los organice, esto fue útil para entender las relaciones entre los conceptos y además para verificar si faltaban conceptos o si habían algunos que no eran relevantes en el desarrollo de su tarea.

5 CONCEPTUALIZACIÓN

La conceptualización consiste en el entendimiento del dominio del problema y la terminología empleada. Se desarrolla en dos etapas. La primera etapa es una actividad de análisis donde se detectan los conocimientos estratégicos, tácticos y factuales. La segunda es un trabajo de síntesis donde los conocimientos detectados en la etapa anterior pasan a formar parte, en mayor o menor medida, de los modelos estático y dinámico, que se integran en el mapa de conocimientos.

El primer paso de la etapa de conceptualización consistió en organizar los conocimientos adquiridos sobre los distintos tipos de máquinas térmicas, su construcción, dispositivos y demás, a fin de identificar conceptos y definir sus propiedades, registrar sus atributos y valores asociados e identificar las relaciones entre conceptos. Se crearon distintos documentos que formaron parte del modelo estático:

- *Glosario de términos* donde se detallo el significado de los términos empleados por el experto
- *Diccionario de conceptos* donde se identificaron los conceptos funcionales de mayor nivel, detallando su utilidad, sinónimos, acrónimos, los atributos que lo definen y la derivación de los datos.
- *Tabla concepto-atributos-valores* en la que se registraron los atributos propios de cada concepto requerido para el modelo de la tarea del experto, detallando para cada atributo el valor o valores correspondientes.

Luego de identificados los conceptos, sus atributos y las relaciones entre ellos, se continuó con el segundo paso, la identificación de los conocimientos estratégicos, tácticos y fácticos.

Los conocimientos estratégicos especifican qué hacer, dónde y porqué hacerlo, es decir, fijan la secuencia de pasos que el SE deberá seguir para ejecutar su tarea. Para ello se identificaron los pasos modulares de alto nivel que el SE ejecutará, el orden y las condiciones en que deberán ejecutarse esos pasos. Luego se dividieron los pasos de alto nivel en tantos subpasos como fue necesario. Finalmente se describió la secuencia de entrada, el modo de razonamiento y las acciones de salida que el SE debe hacer en cada uno de los subpasos del nivel más bajo. En la Figura 2 puede observarse el árbol de descomposición funcional correspondiente a la tarea de determinar la factibilidad de funcionamiento de una máquina térmica.

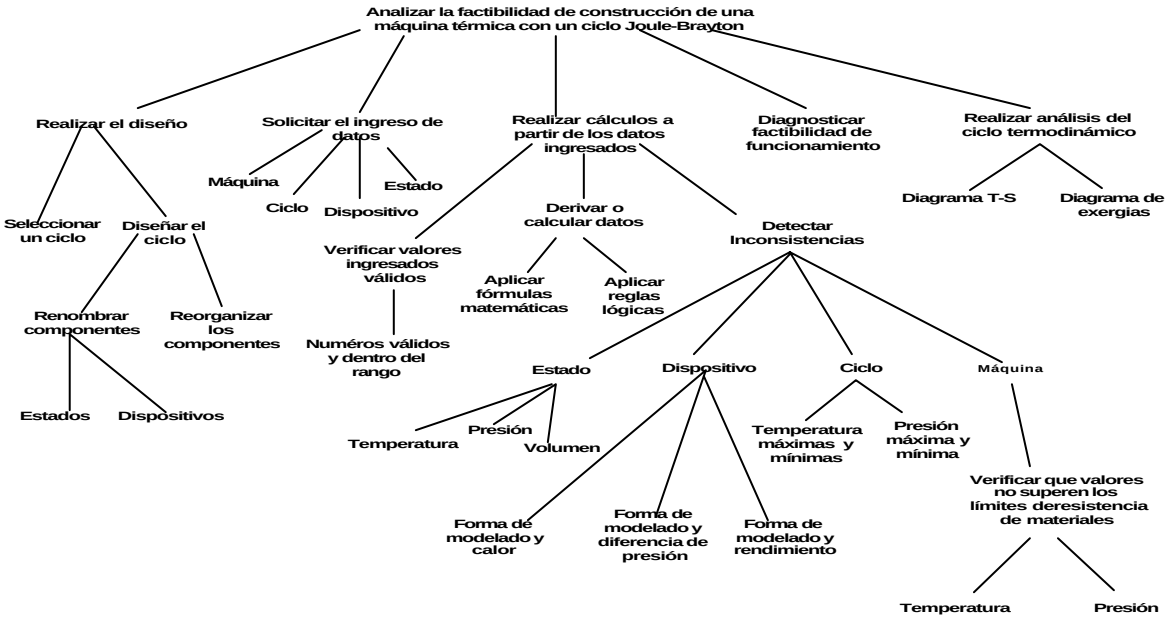


Figura 2: Árbol de descomposición funcional

Continuando con el análisis de los conocimientos se buscaron las inferencias y las incertidumbres que componen los conocimientos tácticos en la creación de un modelo conceptual. Estos conocimientos tácticos del experto especifican cómo el SE puede usar los hechos conocidos e hipótesis del caso para obtener nuevos hechos tanto en situaciones deterministas como en situaciones de incertidumbre. Para ello se analizaron los conocimientos dentro del contexto la tarea que el sistema ejecutará. Con éste análisis se crearon definiciones detalladas de cada paso de razonamiento que deberá ejecutar el SE. Esos pasos son: obtención de conclusiones, reacciones ante nueva información, descripción de la estructura de inferencias y comprobación. Se emplearon seudoreglas para organizar e ilustrar los conocimientos tácticos educidos y tablas para representar cada una de las fórmulas empleadas por el experto para realizar los cálculos necesarios. El SE usará, de un conjunto de fórmulas, las que crea conveniente según los atributos o variables con los que cuente en diferentes circunstancias.
Los conocimientos fácticos del experto contienen información que el sistema conocerá con anterioridad acerca del área de la aplicación, así como información que el sistema obtendrá acerca del caso específico al ejecutar su tarea. Para ello se recopiló y organizó la información acerca de cada atributo general, luego se los clasificó según su importancia para la aplicación y finalmente se buscaron hechos y relaciones entre distintos conceptos independientes de cualquier caso específico.
Finalizada la identificación de los distintos conocimientos se prosiguió con el tercer paso para producir la conceptualización. Se construyó el modelo dinámico o de proceso que lleva a cabo el experto y se comprobó que no hay demasiados errores ni olvidos. Para determinar si un diseño termodinámico es o no factible de llevar a la práctica, el experto realiza los procesos: diseñar la máquina térmica, solicitar datos de las propiedades del ciclo, estados, dispositivos, determinar la factibilidad de la máquina construida y hacer análisis con los datos obtenidos por medio de la construcción de diagramas.
Los modelos estático y dinámico se integran formando el Mapa de Conocimientos. El último paso consistió en la producción del Mapa de Conocimientos que representa el proceso de inferir valores de los atributos. El mismo puede apreciarse en la Figura 3.

6 FORMALIZACIÓN

En el proceso de desarrollo de software existen dos tipos de actividad de modelado. La primera tiene como objetivo crear modelos conceptuales, es decir una representación de los conocimientos del experto externa a la computadora. Los cuales se realizaron en la etapa de conceptualización.
La segunda tiene por objeto crear modelos formales, representaciones semi-computables de los conocimientos y conductas del experto. Para que este modelo formal sea operativo necesita de una base de conocimientos, un motor de inferencias y estrategias de control.
Existen diferentes formalismos para representar los conocimientos del experto los cuales se adaptan en mayor o menor medida al sistema a construir. A continuación se describen los formalismos empleados.

6.1. Sistema de Producción

Un sistema de producción utiliza las implicaciones como base de su representación y su arquitectura esta formada por tres elementos: Base de Hechos (BH) o memoria de trabajo, Base de Reglas o producciones (BR) y una Estrategia de Control (EC). La BH y la BR forman la base de conocimientos del sistema.

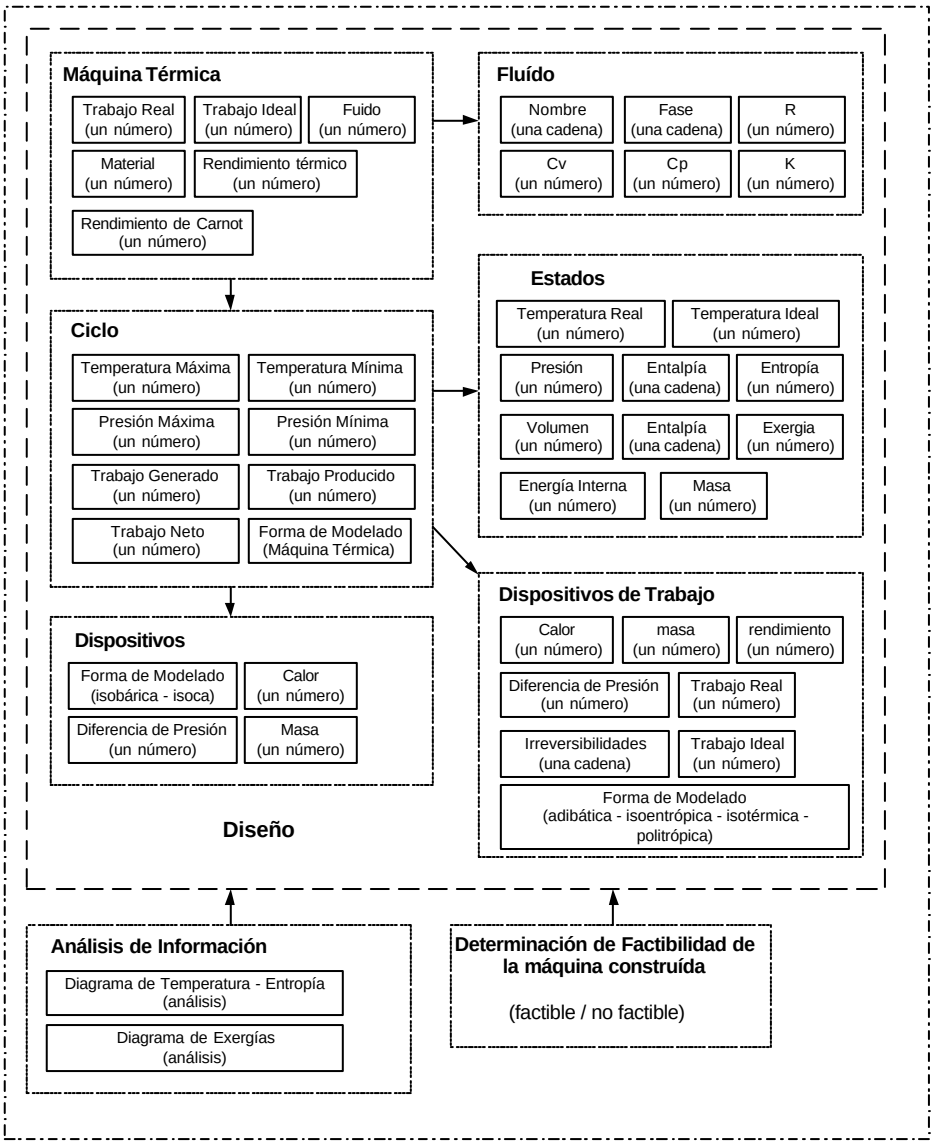


Figura 3: Mapa de Conocimientos

La BH almacena el estado actual del problema, información sobre la tarea y las metas a alcanzar. La BR, formada por un conjunto de reglas, presenta la forma *si condiciones entonces acciones*.

La parte *si*, el antecedente de la regla, representa una lista de cosas a ser verificadas y la parte *entonces*, el consecuente, un conjunto de acciones a realizar sobre la BH. Finalmente, la EC examina la BH y determina la regla que se dispara, encadenando así la regla en unos ciclos de funcionamiento. Cada ejecución de reglas modifica la BH.

Las reglas se construyeron tomando como base el ciclo Joule-Brayton que se muestra en la Figura 1. Algunas propiedades, ya sean del ciclo, del fluido, de los estados del fluido o de los dispositivos, pueden encontrarse en uno de los siguientes estados: el valor de la propiedad no ha sido ingresado o calculado (*N*), ha sido calculado o derivado a partir de otros valores (*C*), fue ingresado por el usuario (*I*). A modo de ejemplo se muestran algunas de las reglas con respecto a la forma de modelado de los dispositivos, en la Figura 4

```
REGLA: "FORMA DE MODELADO ISOENTRÓPICA COMPRESOR"
SI     compresor.formaModelado == "Isoentrópica" y compresor.rendimiento != 1 y
       compresor.rendimientoE !='N'
ENTONCES
       Mensaje de Inconsistencia="Forma de Modelado Isoentrópica en compresor y
       rendimiento distinto de 1"

REGLA: "FORMA DE MODELADO ADIABÁTICA COMPRESOR"
SI     compresor.formaModelado=="Adiabática" y compresor.calor !=0
ENTONCES
       Mensaje de Inconsistencia="Forma de Modelado Adiabática y Calor distinto de cero"
```

Figura 4: reglas con respecto a la forma de modelado

6.2. Marcos

Para los conocimientos del dominio que están organizados en base a conceptos se empleó el formalismo marcos. Los marcos organizan los conocimientos del dominio en árboles construidos por especialización de conceptos generales en conceptos más específicos.
En la Figura 5 se muestran los marcos que representan conceptos, las relaciones que expresan dependencias entre ellos y las propiedades que los describen. Los valores de cada atributo correspondiente a las propiedades de cada marco clase se definieron a través de facetas que expresan de múltiples formas los valores con los que se puede rellenar cada propiedad.

7 IMPLEMENTACIÓN

La representación externa de los conocimientos, obtenidos en la conceptualización, es independiente del entorno en el que va a implementarse el SE. La implementación de sistemas basados en conocimiento exige una herramienta de desarrollo que proporcione los formalismos de representación en los cuales puede codificarse la base de conocimientos y los mecanismos de inferencia y control. El problema planteado conduce mayoritariamente a un enfoque basado en reglas. En esta etapa se procedió a construir las reglas necesarias para implementar el motor de reglas lógico.
Las reglas son básicamente de dos tipos, matemáticas y físicas. Las reglas matemáticas se refieren a las fórmulas que se utilizan para resolver el problema, como ser fórmulas que calculen la temperatura real e ideal en cada estado, presión, entalpía, exergía, etc. Las reglas físicas corresponden a distintos controles para que el sistema sea coherente con la realidad, como ser que la temperatura de entrada debe ser menor a la de salida en la cámara de combustión, restricciones de resistencia de los materiales, de forma de modelado, etc.
Se optó por seleccionar una herramienta que aporte el formalismo de reglas y su motor de inferencias asociado. El sistema experto se implementó con el lenguaje de programación Java siguiendo la tecnología orientada a objetos (OO). Para esto se indagó en distintas alternativas de motores de reglas lógicas que fueran compatibles con la programación OO. Entre las alternativas más viables se seleccionó el motor de reglas Drools, el cual compatibiliza con Java.
En la Figura 6 se muestra la interfaz gráfica del sistema. Como puede observarse, la ventana principal cuenta en la parte superior con una barra de menú, donde se encuentran los menús: *Archivo*, *Ciclo* para crear un nuevo ciclo (Joule-Brayton), *Herramienta* para acceder a conversores de unidades para temperatura y presión y *Ayuda*.
En el sector izquierdo de la ventana podemos ver un área de gráficos, donde se muestra, en la primera solapa, el diseño gráfico del ciclo Joule-Brayton. En la segunda solapa, el análisis gráfico de exergías (recuadro ubicado en la esquina inferior derecha de la Figura 6 y en la última solapa el diagrama T-S (esquina superior derecha) que relaciona temperatura y entropía.

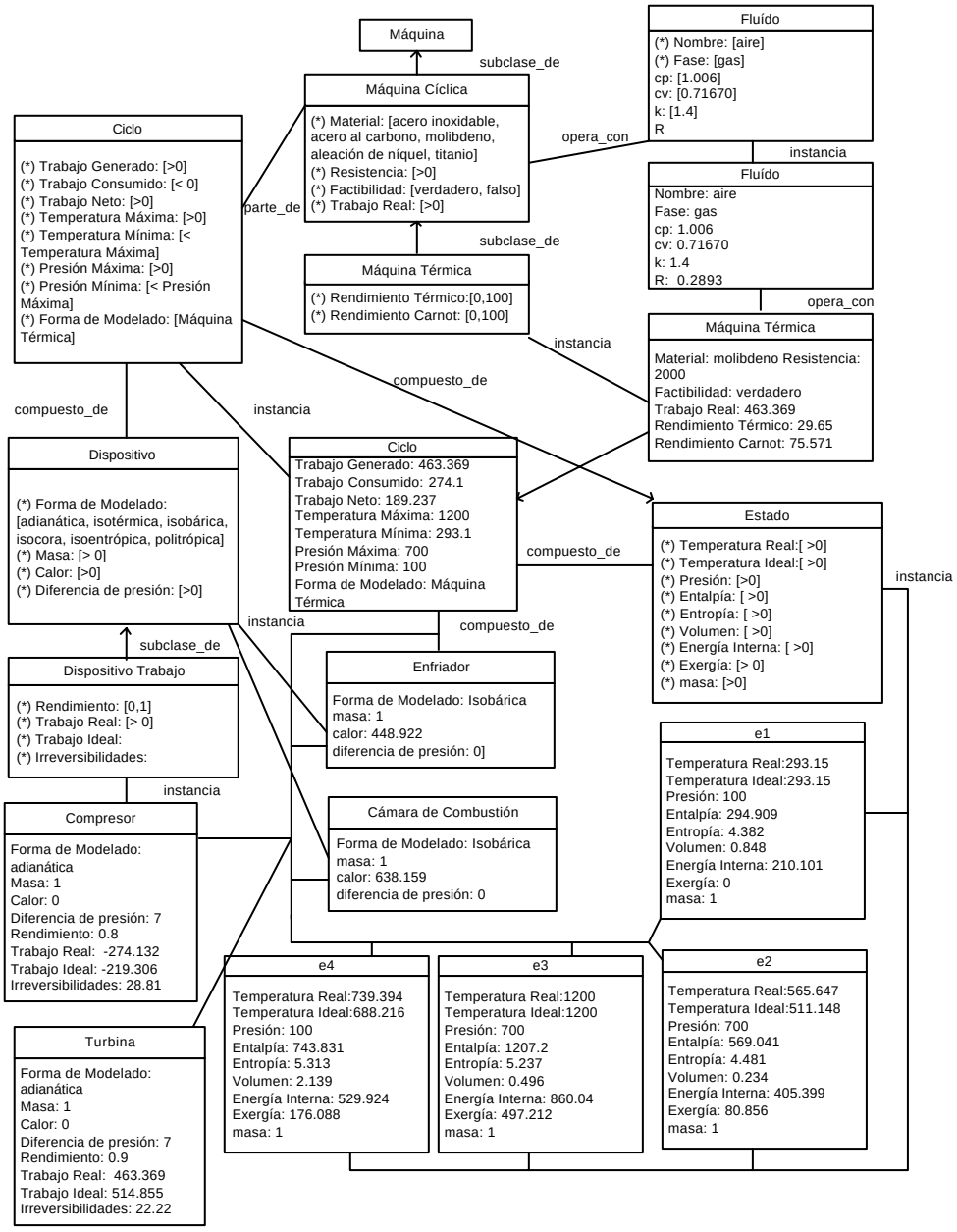


Figura 5: Diagrama de Marcos

A la derecha del área de gráficos se encuentra el panel de propiedades, donde se muestra los valores de las propiedades de estados, dispositivos, ciclos y máquina. Se ingresan valores conocidos relacionados al problema a resolver y se muestran los valores calculados por el sistema. En esta lista de propiedades los colores indican si el valor de la propiedad fue ingresado (verde), calculado por el sistema (azul) o con valor nulo (negro).

En la parte inferior, en la ventana principal, se encuentra el área de mensajes donde se muestran las operaciones que va realizando el sistema. Este panel permite al usuario realizar un seguimiento sobre los valores que las variables van tomando, pudiendo identificar cuales fueron ingresadas, cuales fueron calculadas y también los cálculos que le dieron origen a los resultados. Otra información que se puede analizar en esta sección, son las modificaciones hechas a las propiedades del sistema y los mensajes sobre advertencias, inconsistencias, errores, omisiones y diagnóstico.

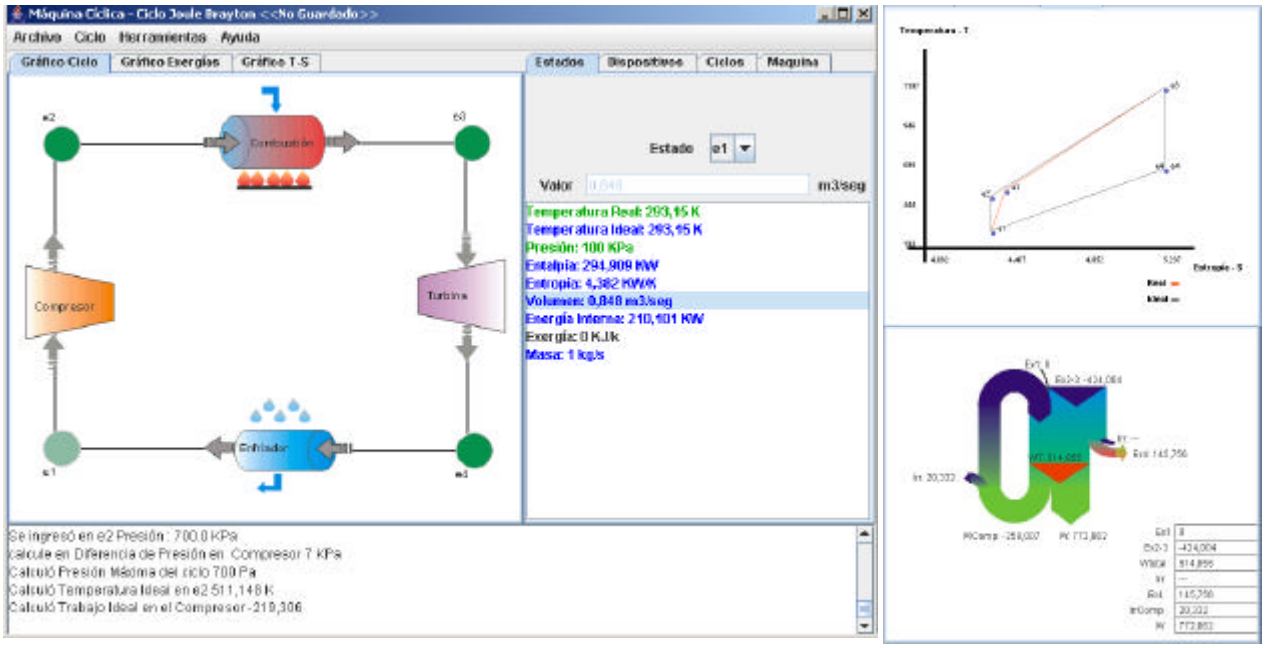


Figura 6: Interfase gráfica del sistema

8 EVALUACION

Durante todo el proceso de construcción del sistema se llevó a cabo, en paralelo, una tarea de evaluación en cada fase del proceso de desarrollo. La evaluación engloba dos acciones, proceso de examen y proceso de juicio. Estas dos acciones se realizaron con mayor profundidad en la etapa final de construcción del sistema. Por lo tanto, el primer paso en el proceso final de verificación de un sistema basado en conocimiento se realiza sobre los modelos o parte de los modelos conceptuales, formales y computables. El objetivo de la verificación, en este punto, es determinar si el modelo sigue las reglas sintácticas del paradigma de representación en el que esta expresado.

Evaluar la sintaxis del modelo obliga a que la verificación sea realizada, en un principio, por los mismos desarrolladores. Para llevar adelante esta tarea se recolectaron más de diez casos de prueba, problemas de termodinámica con sus respectivos resultados, se los ingresó en el prototipo y se contrastó los resultados obtenidos por el sistema con los problemas resueltos. Esta evaluación ayudó a detectar fallas e inconsistencias tanto en reglas matemáticas como físicas, fallas en la estructura del sistema y la coordinación entre los distintos elementos.

En la evaluación de la usabilidad del sistema, el evaluador debe ser el propio usuario, dado que la usabilidad es un criterio subjetivo. Se consideraron dos tipos de usuarios: el experto y el que tiene algunos conocimientos de termodinámica. Para poder llevar adelante de forma correcta la evaluación se planificó un escenario, una proyección de las tareas de prueba que van a realizarse. Esta planificación sirve para parametrizar el estudio.

Se decidió que el propio experto sea el primer evaluador ya que posee amplios conocimientos en el área y conoce las limitaciones del prototipo y la estructura del sistema.

El resultado de aplicar los test realizados al experto fue la redistribución de algunas propiedades, por ejemplo la temperatura y presión ambiente pasaron de estar en el panel ciclo al panel de máquina. También se corrigieron las unidades en la que están expresados algunos valores y el texto de algunos mensajes de error e inconsistencias, entre otros.

Para hacer más exacta la evaluación de usabilidad se eligió como usuario representativo a un estudiante avanzado en la carrera de ingeniería electromecánica. Luego de aplicar el test a este nuevo usuario, como resultado surgió el cambio del texto de algunos mensajes, sobre todo de

inconsistencia, de manera que sean más descriptivos y aporten información más específica sobre el problema al cuál se refieren. Así como también la redistribución de ítems en el menú.

9 CONCLUSIONES Y TRABAJO FUTURO

El presente trabajo constituye un aporte original al proceso de enseñanza-aprendizaje de ciclos termodinámicos. Sistematiza y documenta los conocimientos expertos requeridos para determinar la factibilidad de funcionamiento de una máquina térmica, a través del análisis del diseño de un ciclo de potencia, el ciclo Joule Brayton. Aplica un marco metodológico, a través de la metodología IDEAL, asegurando el desarrollo y posterior crecimiento del Sistema Experto.
En las primeras etapas de testeo, se pudo comprobar que los tiempos de cálculo, de análisis de factibilidad de funcionamiento y de modificación y recálculo de los problemas termodinámicos planteados eran considerablemente inferiores al consumido para la resolución manual. Por lo tanto esto permitirá a los estudiantes enfocarse en los conceptos termodinámicos y optimizar el rendimiento de sus diseños, como se proponía en los objetivos del trabajo.
A futuro, se pretende extender la funcionalidad del sistema de modo que abarque diferentes diseños de ciclos como ser, ciclos abiertos, ciclos Otto, Diesel, Rankine y diseños creados por el usuario, así como también que opere con diferentes tipos de fluidos y trabaje con otras máquinas como bombas de calor y máquinas frigoríficas.
En lo referente a aprendizaje, se espera continuar empleando el sistema en las clases prácticas de la cátedra Termodinámica de la Facultad de Ingeniería de la Universidad Nacional de La Pampa con un grupo "piloto" de alumnos y como estrategia de enseñanza.

REFERENCIAS

[1] Cengel Y. y Boles M. *Termodinámica,* Mc Graw Hill, quinta edición, 2006.

[2] Forbus, D.; Whalley, P. Using qualitative physics to build articulate software for thermodynamics education. *Proceeding of IAAA-94*. Pág.1175-1182. 1994.

[3] Gómez A., Juristo N., Montes C. y Pazos J. *Ingeniería del Conocimiento.* Editorial Centros de Estudios Ramón Areces. 1997.

[4] Gago L., Stark N. Experiencia de Aplicación de las TIC's, en la Enseñanza de las Ciencias, mediante el uso de Sistemas Expertos. *Libro de trabajos del I Congreso en Tecnologías de la Información y la Comunicación (TIC's) en la Enseñanza de las Ciencias*. Pág. 170 a 179. 2005.

[5] Sierra, E., Hossian, A. y García-Martínez, R. 2003. Sistemas Expertos que Recomiendan Estrategias de Instrucción. Un Modelo para su Desarrollo. *Revista Latinoamericana de Tecnología Educativa*. 1(1): 19-30. 2003.

[6] Sierra, E., Hossian, A., García-Martínez, R. y Marino, P.2005. Sistema Experto para Control Inteligente de las Variables Ambientales de un Edificio Energéticamente Eficiente. *Proceedings de la XI Reunión de Trabajo en Procesamiento de la Información y Control*. Universidad Nacional de Río Cuarto. Pág. 446-452.

[7] Bermejo, F., Britos, P., Rossi, B y García Martínez, R. 2002. Sistema de Asistencia para la Configuración de Ventiladores OAF en Neonatolgía. *Revista del Instituto Tecnológico de Buenos Aires*. 28: 24-68. 2002.

Sistemas Tutores Inteligentes: El submódulo de Lenguaje Natural

Zulma Cataldi,
LIEMA - Laboratorio de Informática Educativa y Medios Audiovisuales. Facultad de Ingeniería. UBA.
Facultad Regional Buenos Aires. Universidad Tecnológica Nacional..
liema@fi.uba.ar

Fernando Salgueiro,
LIEMA - Laboratorio de Informática Educativa y Medios Audiovisuales. Facultad de Ingeniería. UBA.
fsalgueiro@fi.uba.ar

Fernando Javier Lage[1]
LIEMA - Laboratorio de Informática Educativa y Medios Audiovisuales. Facultad de Ingeniería. UBA.
Facultad Regional Buenos Aires. Universidad Tecnológica Nacional.
flage@fi.uba.ar

Abstract

In previous publications the low yield of the students in its partial and final evaluations had been detected. By that reason thought about supporting the actual classes with intelligent tutoring. The basis for the design of intelligent tutorial system beginning by the redesign of the modules that compose it were done. Now, one looks for to incorporate a natural language module that is independent of the rest of components taking like example the contents of basic programming.

Keywords: Intelligent Tutoring Systems, natural language.

Resumen

En publicaciones previas se había detectado el bajo rendimiento de los estudiantes en sus evaluaciones parciales y finales. Por ese motivo se pensó en apoyar las clases presenciales con tutorizado inteligente. En las primeras publicaciones se sentaron las bases para el diseño del sistema tutor inteligente comenzando por el rediseño de los módulos que componen el nuevo modelo. Ahora, se busca incorporar un módulo de lenguaje natural que sea independiente del resto tomando como ejemplo los contenidos de programación básica.

Palabras Clave: Sistemas Tutores Inteligentes, lenguaje natural.

[1] Esta comunicación ha sido desarrollada en el marco del Convenio FI-UBA y UTN-FRBA. Proyecto *Modelado del tutor basado en redes neuronales para un Sistema Tutor Inteligente* del Programa de Incentivos 2007-2008.

1. INTRODUCCIÓN

Se busca definir las interfaces para poder incorporar un submódulo de lenguaje natural a la estructura del módulo del tutor en un STI. Para ello, se utilizarán los lineamientos de ya presentado en trabajo previos [1] y los aportes de Di Eugenio [2], quien plantea que una de las claves para solucionar el abismo actual que existe entre los sistemas tutores inteligentes de esta generación y sus contrapartes humanas reside en la forma en que el primero interactúa con el humano por lo que realiza un análisis completo de la mejora que puede garantizar el uso de lenguaje natural en los STI. Este tema fue tratado por otros autores, como Evens *et al.* [3] quien aportó el resultado de su investigación al proyecto del tutor *CircSiM*. Freedman [4] y Graesser *et al.* [5] entre otros, realizaron notables aportes acerca de los beneficios que conlleva la incorporación de lenguaje natural en los STI. Este módulo está relacionado con el planificador de la lección y genera las salidas del sistema con lo cual se consigue tener un sistema tutor inteligente con capacidades de lenguaje natural. Pero, estas funciones están intrínsecamente enlazadas con el sistema del planificador y no se pueden ver cómo módulos realmente separados. En este caso se propone, el análisis del módulo de lenguaje natural como una *"caja negra",* a fin de definir las funcionalidades y sus interfaces con los demás módulos para hacerlo completamente independiente de éstos. Algunos tutores que en la actualidad utilizan el lenguaje natural como en forma integrada son: *Circle* y *AutoTutor System* [5] y se están desarrollando mejoras en tutores más antiguos para brindarles las capacidades del lenguaje natural [5][6][7].

2. FUNDAMENTACION

Se busca obtener un submódulo independiente de lenguaje natural con lo que los primeros prototipos tendrán capacidades limitadas pero en la medida que el desarrollo avance se podrán obtener módulos más potentes y completos con solo modificar el submódulo en cuestión.
La modularización permite comparar el desempeño de cada uno de los módulos implementados por separado, con lo que el módulo de lenguaje natural se puede cambiar para idiomas diferentes al realizar la sesión de enseñanza, logrando de este modo un tutor multilenguaje, sin tener que modificar el resto de los módulos, solo agregando la información en el módulo de conocimientos en los distintos idiomas. En el modelo más tradicional, muchas de las frases son de tipo "*hardcoded*"[2] es decir pre-armadas dentro del planeador, quien es el encargado de producir la salida del sistema, tal como en las primeras versiones del tutor *CircSim* [8].
Si se analiza el módulo de lenguaje natural como una caja blanca, se encontrará que el proceso más complejo que debe realizar es el denominado: "*alegación de sentencias*" de Reiter y Dale [9] quienes lo definen como el proceso de agrupación de sentencias a fin de definir oraciones. Esto se puede lograr obteniendo información acerca de la sintaxis del lenguaje humano de una base de datos interna, de la que se obtendrá el vocabulario para armar las oraciones, el conjunto de reglas que determinan las combinaciones que están permitidas en el lenguaje, las reglas de agregación, etc. Estos aspectos fueron estudiados por, Huang y Fiedler, Shaw [10] y Scott y Sieckenius de Souza [11] entre otros. Independientemente del algoritmo que se utilice para la construcción lingüística, la independencia del módulo y el uso de la interface permitirá que el STI pueda obtener la capacidad de lenguaje natural, desde la simple generación de frases aisladas, hasta el armado de párrafos relacionados entre si. Debido a ello, se definirá primero la forma de representación de la interface entre los submódulos del módulo tutor y el que corresponde al lenguaje natural. Esta consistirá en una serie de funciones básicas con sus parámetros, que permitirán realizar la interacción con el usuario final del sistema en ambos sentidos al submódulo de lenguaje natural, es decir, desde el sistema hacia el usuario (como el *output* o salida del sistema) ó desde el usuario que

[2] Vocablo que se usa Para de definir valores o parámetros que se encuentran fijos en el código fuente y no pueden ser cambiados una vez compilado el programa, restándole así flexibilidad.

responde a alguna de las preguntas del sistema (como uno de los múltiples *inputs* ó entradas del sistema). Para ello, se deberán definir previamente los objetivos del módulo de lenguaje natural; que son:

- *Realizar la salida de datos de manera similar se esperaría de un tutor humano*: La salida de los datos se puede realizar de varias maneras, siendo éstas completamente independientes de los datos que pertenecen a la interface, que es la encargada de mostrar los contenidos multimediales de las lecciones ó sesiones de tutelado y de presentar una pantalla homogénea para navegar a través del sistema. Por ejemplo, la salida puede ser en forma de texto coloquial en el idioma nativo del usuario, es decir podría ser español incluyendo los términos técnicos asociados al vocabulario correspondiente a un alumno de nivel inicial que va a realizar un primer curso o en forma de voz, aplicando las mismas consideraciones que para el texto. Estas dos opciones no son mutuamente excluyentes y podrían utilizarse ambas, por medio de aplicaciones de conversión de terceros tales como Microsoft Agent[3].
- *Entregar los datos procesados a la interface:* Es decir, limpios de todas construcciones gramaticales utilizadas por el alumno como medio de construcción de respuestas u otras preguntas que se le deseen hacer al sistema. Se busca que el submódulo de lenguaje natural se encargue de procesar la entrada de los datos de la interface, que representa la interacción con el usuario obteniendo de ésta todos los conceptos filtrados, es decir, libres de las construcciones gramaticales que los soporta en el lenguaje natural. Estos datos luego serán enviados desde la interface para su procesamiento hacia los otros submódulos.
- *Funcionar como entrada de los datos también en lenguaje natural:* Está pensado para que el alumno pueda desenvolverse con mayor facilidad al responder los cuestionarios del sistema ó ante cualquier otra exposición de datos que requiera una interacción del alumno, siempre guiada por parte del sistema. Este sistema puede funcionar tanto en forma escrita como oral, utilizando las herramientas adecuadas[4].

Con estas tres funciones básicas y una vez definida la interface, se pueden establecer las bases para construir un submódulo de lenguaje natural que sea independiente del resto del tutor. Es más, se puede abstraer aún más la estructura para lograr un submódulo completamente separado, si se desea hacer un sistema basado en tecnologías del tipo *Web services*, en las que los módulos principales se encuentren en un servidor remoto y tanto la interface como el submódulo de lenguaje natural se pueden encontrar en el cliente. De esta manera, el proceso de transformación de las entradas, se distribuirá en el cliente y se minimizará la cantidad de información a ser transferida por la red, mejorando así el rendimiento global de todo el sistema.

3. LOS SISTEMAS TUTORES INTELIGENTES

Los sistemas tutores inteligentes (STI) comenzaron a desarrollarse en los años 80 y fueron diseñados con la idea de impartir conocimiento con base en alguna forma de inteligencia para guiar al estudiante en el proceso de aprendizaje [12][13]. Su propósito es presentar un comportamiento similar al de un tutor humano, que se adapte a las necesidades del estudiante, identificando la forma en que el mismo resuelve un problema para poder brindarle ayuda cuando cometa errores. Un tutor inteligente, por lo tanto: *"es un sistema de software que utiliza técnicas de inteligencia artificial (IA) para representar el conocimiento e interactúa con los estudiantes para enseñárselo"* [14]. Wolf [15] define los STI como: *"sistemas que modelan la enseñanza, el aprendizaje, la comunicación y el dominio del conocimiento del especialista y el entendimiento del estudiante*

[3] *Microsoft Agent* es una tecnología que presenta una interface conversacional que puede, desde aceptar comandos verbales, hasta hablar (por medio de motores de voz en varios idiomas o audio grabado).Puede ser embebido dentro de páginas Web, aplicaciones, etc. para mejorar y humanizar la interacción de éstas con el usuario (Información disponible en su sitio Web: http://www.microsoft.com/msagent/). Los motores de voz pueden ser los provistos por Microsoft o por terceros, de los cuales vale la pena resaltar la empresa *lernout & hauspie*, disponible en su sitio Web: http://www.lhs.com/).

[4] Microsoft Agent ó alguna otra similar, como Dragon Naturally Speaking Voice Recognition de DragonTalk Información disponible en su sitio Web: http://www.dragontalk.com/. Consultada el 24/07/2007

sobre ese dominio". Giraffa [16] los delimita como: *"un sistema que incorpora técnicas de IA (Inteligencia Artificial) a fin de crear un ambiente que considere los diversos estilos cognitivos de los alumnos que utilizan el programa"*. En la Figura 1 se puede ver la estructura general de un STI donde el sub-módulo generador de lenguaje natural se encuentra dentro del módulo del tutor y es el encargado de interactuar directamente con la interfaz, transmitiendo los conocimientos, las respuestas y las repreguntas en forma de lenguaje (hablado, escrito u de otra manera) para que el estudiante se sienta mas cómodo utilizando el sistema.

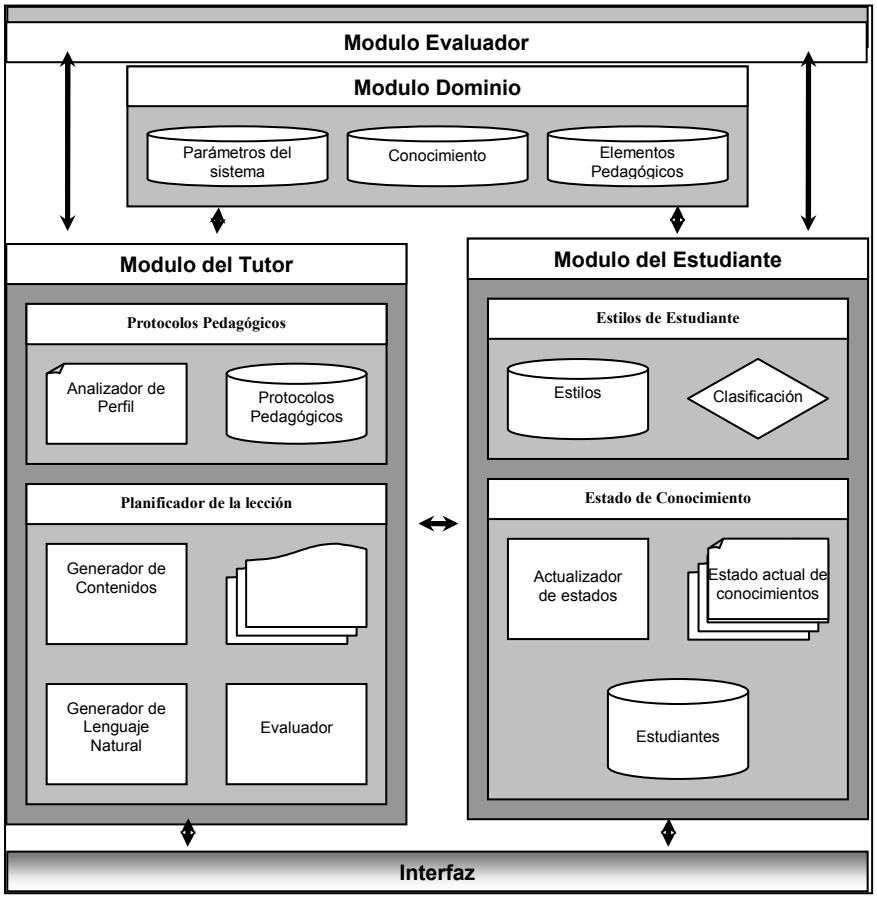


Figura 1: Modelo detallado de un Sistema Tutor Inteligente

Los STI permiten la emulación de un tutor humano para determinar *qué enseñar, cómo enseñar y a quién enseñar* a través de un *módulo del dominio*: que define el dominio del conocimiento, *un módulo del estudiante*: que es capaz de definir el conocimiento del estudiante en cada punto durante la sesión de trabajo, un *módulo del tutor*: que genera las interacciones de aprendizaje basadas en las discrepancias entre el especialista y el estudiante y finalmente *la interface*[1] con el usuario: que permite la interacción del estudiante con un STI de una manera eficiente aplicando el conocimiento sobre *cómo presentar* los contenidos.

4. LAS INTERFACES DEL SUBMÓDULO DE LENGUAJE NATURAL

Para que el sistema funcione se definirá analizarán los tipos de interacciones posibles con el módulo de lenguaje natural:

- *Interacciones de tipo informativas:* A través de estas interacciones, el módulo tutor podrá agregar datos de interés, subrayar las relaciones entre los conceptos y aclarar las dudas particulares a lo largo del dominio en este caso de la programación básica. Este tipo de interacciones surgen cuando el módulo tutor está realizando una sesión de tutelado con el usuario pero no durante los interrogatorios.
- *Interacciones de tipo pregunta:* Se deben utilizar cuando el sistema requiere hacer una pregunta directamente al usuario, relacionada con algún concepto, para verificar que el usuario puede manejarlo o para agregarlo a la pila[5] de temas desconocidos cuando éste falla en responder a la pregunta de forma satisfactoria.

[5] Una pila (stack en inglés) es una estructura de datos de tipo LIFO (del inglés Last In First Out, último en entrar, primero en salir)

- *Interacciones de tipo respuesta:* Una vez planteada la pregunta por el sistema, se pueden presentar varias situaciones de respuesta por parte del alumno, como las planteadas por Yujian [17], y otras nuevas que para poder ser implementadas en el modelo propuesto dependiendo del tipo de respuesta suministrada por el alumno, el sistema responderá de la siguiente manera:
  - *Si la respuesta es Correcta:* El sistema se encargará de hacerle saber al usuario que su respuesta es correcta, y actualizará la pila de objetivos de la lección con respecto al nuevo estado. Este es el caso más simple, ya que el sistema sólo debe comunicarle al usuario que está en lo correcto.
  - *Si la respuesta es "Near Miss"* [18]: es decir, cundo la respuesta no es la deseada pero es pedagógicamente útil. Un ejemplo puede ser la siguiente interacción:

```
P: ¿Cuáles son los tipos ordinales en Pascal?
R: Los numéricos son tipos ordinales.
```

    La respuesta es incorrecta, porque si bien los tipos enteros (*integer, byte, longint, etc.*) son ordinales, los numéricos *reales* no lo son, por lo tanto el sistema deberá decidir si debe explicarle esto al alumno, si debe utilizar alguna pista para ayudarlo a dilucidar la respuesta correcta ó si debe darle la respuesta directamente para que éste pueda reflexionar e incorporar el concepto que se intenta explicar.
  - *Si la respuesta es "no sé":* Dependiendo del caso puede pasar a una interacción del tipo pista, pero si el error es muy grave ó conceptual esto puede derivar hacia una respuesta interacción del tipo informativa. En ambos casos no se involucra una interacción de tipo respuesta.
  - *Si la respuesta es un error conceptual* [15]: También se puede optar por una respuesta interacción del tipo informativa pero que no involucra una interacción de tipo respuesta.
  - *Si la respuesta es de otro tipo:* El sistema deberá responderla con una interacción del tipo informativa, para simplificar el esquema.

En cualquiera de los casos el sistema deberá poder armar sus respuestas clasificando forma en la que tratará cada uno de los tipos de respuesta.

5. ANÁLISIS DE LOS TIPOS DE INTERACCIONES

Existen diversas categorías de interacciones:

Interacciones de tipo Pista: Con base en el modelo de pistas propuesto por Hume *et al.* [7] existen distintas formas de presentar las pistas: a través de pregunta, como afirmaciones, relacionándolas con otros conceptos, aunque todas las formas para presentar una pista, dan la misma información, independiente del modo en que se presente. Un ejemplo de este caso puede ser el siguiente:

```
P: ¿Cuáles son los tipos ordinales en Pascal?
R: Los numéricos son tipos ordinales.
```

Si el sistema decide dar una pista, podría realizarla de la siguiente manera:

```
Pista: Los números reales son del tipo numérico, pero: ¿son ordinales?
Pista: Piense en los números reales.
Pista: ¿Cuál es la característica fundamental de los ordinales?.
```

Las dos primeras son dos formas de mostrar la misma información: como pregunta y como afirmación. La última, requiere que el planificador de la lección agregue a la pila de objetivos uno nuevo que es: "*Características de los ordinales*", que deberá ser respondido de forma satisfactoria antes de continuar con la pregunta original que era: *¿Cuáles son los tipos ordinales en Pascal?*. Se

dejará al submódulo de lenguaje natural la forma en que presentarán las pistas, es decir, si serán como pregunta o afirmación.

– *Interacciones de tipo Control:* Se utilizan cuando se desea que el sistema presente las respuestas de una manera en particular, de forma forzada como pregunta, de forma forzada como afirmación y proporcionando estructuras de control para que se puedan presentar varias interacciones al mismo tiempo, lo cual lleva a la próxima clasificación:

– *Interacciones de tipo Mixtas:* En este caso el sistema requiere de una mezcla de respuestas para mejorar la interacción con él usuario. Un ejemplo podría ser el siguiente:

```
P: ¿Cuáles son los tipos ordinales en Pascal?
R: Los numéricos son tipos ordinales.
```

La respuesta del sistema puede ser una combinación de dos respuestas: una de tipo interacción informativa, seguida por una interacción de tipo pregunta, de la siguiente manera:

```
Los tipos ordinales se caracterizan en que dado un elemento cualquiera del conjunto, se
puede conocer cuál es su predecesor (salvo del primero) y el sucesor (salvo del
ultimo). Ahora: ¿Todos los tipos numéricos poseen esta característica?.
```

Aquí se ve que el sistema da la definición del concepto y luego se encarga de generar una nueva pregunta, con lo que se tienen dos interacciones, que son parte de la misma respuesta del sistema a la entrada provista por el usuario.

En la Figura 2 se transcribe un ejemplo de interacción entre un alumno y un tutor humano. A partir de una serie de interacciones reales como éstas surge la clasificación de las respuestas del tutor dentro del modelo anteriormente planteado.

```
P: ¿Cuáles son los tipos ordinales en Pascal?
A: Los numéricos son tipos ordinales.
P: No todos los tipos numéricos son ordinales. Puede decirme cuáles son los tipos numéricos que
    conoce?.
A: Los tipos numéricos son Byte, Integer, Longint,  Real, Extended, etc.
P: Si, pero de una forma más teórica, todos los tipos numéricos que usted mencionó pueden agruparse
    en dos categorías.
A: Enteros y Reales.
P: Si. Ahora, retomando la pregunta original, esos dos tipos numéricos son ordinales?.
A: Si.
P: No, los tipos numéricos reales no son ordinales, ¿Sabe usted por qué?
A: No.
P: Piense en lo siguiente: ¿Cuál es la característica fundamental de los ordinales?.
A: Que tienen un orden.
P: Si, Esa afirmación está incompleta. Plantee mejor qué significa que los ordinales tienen un
    orden.
A: Que un elemento es mayor que el otro.
P: Eso no es correcto en todos los casos, puede completar la idea.
A: No, no puedo.
P: Un conjunto ordinal es aquel en el cual el compilador puede identificar el predecesor de un
    elemento iésimo del conjunto (menos el primero) y un sucesor del elemento iésimo (menos del
    último).
P: Entonces, podría decirme por qué los reales no son un conjunto ordinal.
A: Porque dado un elemento, el compilador no puede calcular cuál es el siguiente inmediato.
P: Correcto.
```

Figura 2: Modelo de interacciones entre un estudiante y un tutor humano.

También se deben definir las funciones que caracterizaran a la interface entre el submódulo de lenguaje natural y el resto del módulo del tutor, es decir:

– *Interacciones de tipo informativas*: `I_Information (Tipo, Texto)`

Este tipo de interacción es la más simple de todas y brinda la definición del concepto a explicar desde el módulo del dominio y lo muestra en pantalla. Esta es una función del submódulo de lenguaje natural que permite mostrar el texto que se encuentra en el campo de texto. El parámetro opcional "`Tipo`" surge porque el generador de lenguaje natural no está limitado a producir salidas de manera entendibles sólo en forma de texto (escrito u oral) para el usuario humano. Esta situación se

puede aclarar a través de un ejemplo suponiéndose un STI cuyo dominio de aplicación es la enseñanza de un instrumento musical, por ejemplo: el piano. El Sistema Tutor podría estar intentando mostrar una partitura que pretende que el alumno represente en el instrumento (a través de la salida del planificador de la lección hacia el modulo de lenguaje natural), mientras que el estudiante puede tocar en un instrumento real y el submódulo de lenguaje natural deberá decodificar las notas por tonalidades y entregarlas al planificador codificadas tal como se ve en la Figura 3.

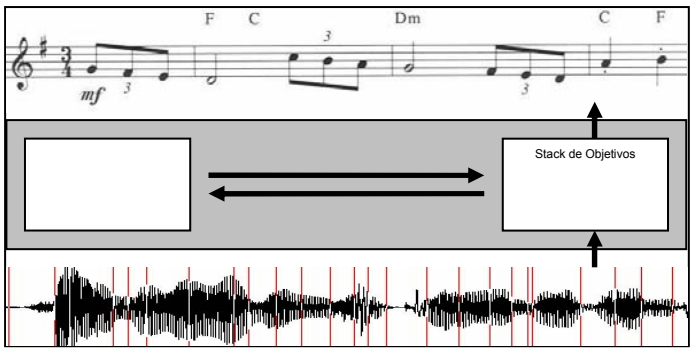


Figura 3: Interacción no convencional para producir salidas que no son texto y procesar entradas que no lo son.[1]

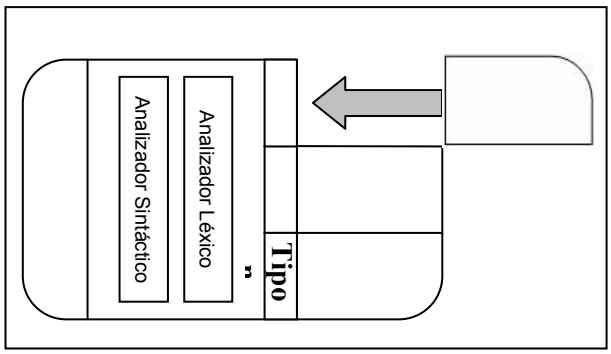


Figura 4: Submódulo de lenguaje natural como un núcleo central de funcionalidades y un conjunto de componentes adicionales.

Este tipo de interacciones específicas para algunos STI, no están limitadas por la definición de la interface, están permitidas y lo único es que se requiere un submódulo de lenguaje natural que las soporte, es decir, que cumpla con las reglas generales de la interface y que luego utilice estos `Tipos` que se dejan sin definición para casos particulares. Esto brinda la flexibilidad para que no se requiera modificar completamente la estructura del submódulo de lenguaje natural en los STI, sino que se pueda optar por una estructura de "*componentes*" para el submódulo de Lenguaje Natural donde cada uno de estos "*componentes*" se referencia con el número que indica el parámetro "`Tipo`" como se muestra en la Figura 4.

- *Interacciones de tipo pregunta:* `I_Question (Tipo, concepto`$_0$`,concepto`$_1$`,…, concepto`$_n$`)`

  Este es un método secundario que permitirá al tutor obtener los resultados de una sesión pedagógica si se utilizan protocolos de tutoría del tipo magistrales, mientras que será el método principal si se utilizan protocolos de tutoría del tipo socrático [1]. En este tipo de interacción se busca que el submódulo de lenguaje natural genere construcciones lingüísticas compatibles con las preguntas. Éstas se podrían categorizar teniendo en cuenta los modelos registrados durante las preguntas posibles a realizar por el sistema en los interrogatorios y las evaluaciones de la asignatura:

  – *Preguntas de conceptos:* Estas preguntas apuntan a que el usuario del sistema entregue la definición de un concepto en particular. Éstas son las preguntas más comunes en las sesiones socráticas y probablemente no se encuentren en las sesiones magistrales. Ejemplos de estas preguntas son las siguientes:

Pregunta real	Primitiva de Interacción
¿Cuáles son los tipos ordinales en Pascal?	I Question ("concepto", "tipos ordinales")
¿Cómo se pueden clasificar los números enteros?	I_Question ("concepto", "clasificación de enteros")

– *Preguntas de relación de conceptos:* Se intenta que el alumno relacione dos conceptos que poseen una o más características en común. Este tipo de preguntas ayuda al planificador como un indicador de que el alumno puede integrar varios conceptos. Son ejemplo de este tipo de preguntas:

Pregunta real	Primitiva de Interacción
¿Qué tienen en común la codificación "módulo	I_Question("relación", "módulo y signo",

y signo" y el binario puro?	"binario")
¿Qué características comparten los Integer, Longint y Byte?	I_Question("concepto","Integer","Longint","Byte")

–*Preguntas de ejercicios:* Con estas preguntas que si bien pueden suponerse como de concepto, se espera del alumno un resultado conciso, pero no necesariamente definiciones, ya que pueden ser resultados numéricos, códigos fuentes, etc. También se pueden enmarcar en esta categoría las preguntas de respuesta múltiple y las de "*verdadero o falso*", que pueden formar parte de exámenes o parte de la lección, según pertenezcan al planificador o no. Algunos ejemplos son:

Pregunta real	Primitiva de Interacción
¿Qué valor representa el número 179 decimal en binario?	I_Question ("ejercicio", "decimal a binario", "179")
¿Cuántas iteraciones máximas debe hacer para ordenar por el método de Burbujeo un vector de 10 posiciones?	I Question ("ejercicio", "ordenamiento", "burbujeo", "iteraciones",)

–*Pregunta de definiciones:* Son aquellas en las que el sistema espera que el alumno proporcione la definición exacta de un concepto y no solo alguna de sus características (para ello, están las preguntas de *concepto*). Esto podría enmarcarse como preguntas de exámenes ó similares para que un tutor luego las analice. Este tutor puede ser un tutor humano: Las preguntas de este tipo quedaran almacenadas en el STI, quien se encarga de procesar las respuestas como si este tipo de preguntas fuera alguna de las anteriores. Ejemplos de éstas serían:

Pregunta real	Primitiva de Interacción
Defina las partes de la CPU	I_Question ("definición", "partes CPU")
¿Qué entiende por recursividad?	I_Question ("definición", "recursividad")

–*Otras preguntas:* En esta categoría se engloban todas las preguntas que no responden a ninguna de las mencionadas anteriormente. Se pueden utilizar para mantener un diálogo fluido entre las distintas preguntas generales del sistema, cuando el planificador agrega a la cola[6] conceptos nuevos a explicar al alumno.

Pregunta real	Primitiva de Interacción
¿Sabe usted por qué?	I Question ("otras", "¿Sabe usted por que?")
Plantee mejor qué significa que los ordinales tienen un orden.	I Question ("otras", "Plantee mejor que significa que los ordinales tienen un orden.")

La clasificación de las preguntas tiene como objetivo facilitar el "*parsing*" (analizador sintáctico) de la respuesta del alumno, es decir, si el sistema tiene que interpretar una respuesta sin conocer el contexto de la pregunta, puede ser mucho más difícil obtener datos concisos. El tipo de pregunta le entrega al submódulo de lenguaje natural el contexto en el cual deberá buscar la respuesta de la próxima respuesta del alumno. En caso de que la pregunta se encuentre dentro de un bloque generado por las interacciones de control (`I_control ("comienzo")` y `I_control ("finalización")`), el submódulo de lenguaje natural deberá mantener el contexto de la última de las `I_Question` que reciba. En la mayoría de los casos generales, es muy poco probable que el planificador entregue más de una pregunta dentro `I_Question` del mismo bloque, ya que esto representaría que el alumno debería responder dos preguntas ó más dentro de la misma respuesta, aunque esta interacción no está prohibida en este protocolo. Un ejemplo podría ser:

Pregunta real	Primitiva de Interacción
P: Un conjunto de elementos Ordinal es aquel que el compilador puede identificar el predecesor de un elemento iésimo del conjunto (menos el primero) y un sucesor del elemento iésimo (menos del último). Entonces, podría decirme por qué los reales no son un conjunto ordinal?.	I_control ("Comienzo") I_Information ("Conjunto ordinal") I_Question ("concepto", "reales", "conjunto ordinal") I_control ("finalización")
Si, Esa afirmación está incompleta. Plantee mejor qué significa que los ordinales tienen un orden?.	I_control ("Comienzo") I response ("Si") I Question ("otras", "Plantee mejor que significa que los ordinales tienen un orden.") I_control ("finalización")

[6] Una cola (queue en inglés) es una estructura de datos de tipo FIFO (del inglés Firs In First Out, primero en entrar, primero en salir)

- *Interacciones de tipo respuesta:* `I_Response (Tipo_Respuesta, texto)`

Simplemente son el resultado de una respuesta del sistema quien logra responder básicamente en forma afirmativa o negativa las interacciones con los usuarios, pero se puede agregar un tipo de respuesta, que se denominará neutra, para representar las situaciones incompletas, donde la respuesta dada es solo una parte de la respuesta esperada por el sistema, pero no es la respuesta completa esperada. En todos los casos, el texto que acompaña a este tipo de interacción es opcional, el planificador puede proveer el resultado exacto que se desea agregar en la respuesta. Esto también puede ser emulado utilizando interacciones de control para generar un bloque, pero independientemente de que se encuentre el parámetro opcional o no, la clasificación de los tipos de respuesta es la siguiente:

–*Positiva*: Cuando el sistema recibe la respuesta que desea para alguno de los conceptos que han sido preguntados y se encuentran en la pila del planificador de la lección. En ese momento, el planificador la retira de la pila, pero este proceso es transparente para la interface.

Pregunta real	Primitiva de Interacción
`Si (el tipo más sencillo de esta categoría)` Es correcto, los enteros son ordinales.	`I_Response ("positiva")` `I_Question ("positiva", "Es correcto, los enteros son oridnales.")`

–*Negativa*: Cuando el sistema recibe una respuesta errónea y no desea generar pistas u otro tipo de ayuda para corregir al usuario, muestra una respuesta del tipo negativa. En la mayoría de los casos, se encontrará dentro de un bloque, ya que el sistema no solo debe limitarse a informar el estado de la respuesta del usuario, sino que deberá modificar los conocimientos del usuario para que este adquiera el concepto, si ésta fuera incorrecta, es decir, por la propia definición de STI que cautele la apropiación de conceptos. El protocolo de la interface es independiente de que la respuesta negativa se encuentre en un bloque ó no. Si no se encontrara en un bloque, el submódulo de lenguaje natural deberá mostrar la respuesta aunque este caso no debería ser la salida del planificador de la lección. Es decir:

Pregunta real	Primitiva de Interacción
`No (el tipo mas sencillo de esta categoría)` Eso no es cierto, no todos los formatos numéricos son ordinales.	`I_Response ("negativa")` `I_Question ("negativa", "Eso no es cierto, no todos los formatos numéricos son ordinales.")`

–*Neutra*: Cuando el sistema responde al usuario de una manera en la que la respuesta a la interacción no es completa, ó la esperada por el sistema. Además del caso de las respuestas incompletas a preguntas respecto de una serie de conceptos, se puede agregar este caso a las situaciones en las que el planificador no puede comprender correctamente la respuesta del usuario y por lo tanto debe reformular la pregunta para que el usuario la entienda correctamente.

- *Interacciones de tipo Pista:* `I_Hint (texto)`

Las interacciones del tipo "*pistas*", son interacciones similares a las respuestas, pero por su valor pedagógico, se ha decidido desglosarlas en una nueva categoría. La categorización utilizada para los distintos tipos de pistas es similar a la que plantea Hume [7] en su tesis doctoral, que engloba a las de Glass [18] y las de Wolf [15] donde el tratamiento de las pistas se puede ver como una combinación de las primitivas básicas planteadas en este modelo. Cada uno de los casos de las pistas se puede aclarar en forma particular:

–*Si la respuesta es del tipo Correcta:* No requiere ninguna aclaración en especial genera una interacción del tipo `I_Response ("positiva")` simple y de la manera explicada anteriormente.

- *Si la respuesta es del tipo "Near Miss*" [18]: Este es el primer tipo de respuesta que puede contener una verdadera interacción del tipo `I_Hint`. El alumno no acertó correctamente la respuesta, pero demuestra tener conocimientos teóricos sobre el tema, el sistema deberá guiarlo como se explicó anteriormente, generando una nueva pregunta orientativa del camino mental a seguir. Analizando este caso como una interacción de tipo mixta, deberá tener una forma como la siguiente:
  - *Si la respuesta es del tipo "no se":* En este caso en particular, el sistema puede optar por seguir entregándole más pistas para que el estudiante logre encontrar la solución por sus propios medios ó puede desistir a las interacciones de tipo pista para entregarle la definición del concepto pedido o directamente la respuesta a la pregunta.
  - *Si la respuesta es del tipo error conceptual* [15]: Si se encuentra con un error conceptual, el planificador debe encargarse de corregirlo, se finalizará con las interacciones de tipo pista y se continuara con interacciones de tipo `I_information`.
  - *Si la respuesta es de otro tipo:* El planificador debe encontrar la solución correcta a una respuesta no categorizada, puede optar por reformular la pregunta original (`I_Question`), por entregarle al usuario estudiante otra pista que lo acerque nuevamente al camino por el cual el STI lo desea guiar (`I_Hint`) ó por último puede desistir y entregarle la definición ó la respuesta a la pregunta que dio origen a esta interacción en primer lugar (`I_information`).

- *Interacciones de tipo Control:* `I_control (Tipo)`

Todas las interacciones de control surgen de la necesidad de poder armar interacciones complejas como las que plantean Reiter y Dale [9] y su propósito principal es el de la construcción lingüística. Varias respuestas pueden estar agrupadas para formar oraciones y estas oraciones pueden unirse luego para formar párrafos. Por la naturaleza de la interface, cada una de las llamadas particulares de estas funciones generará por sí misma una contracción lingüística correspondiente a una oración o más. Pero, una misma respuesta del sistema puede contener una `I_Response` (como resultado directo de la ultima interacción con el usuario), seguido directamente por una `I_Information` (para aclarar algún concepto errado o para ampliar el conocimiento sobre un tema en particular), y por último una `I_Question` para verificar que el usuario ha entendido la nueva información que se le presento y que puede seguir adelante con el proceso de aprendizaje. El ejemplo citado en el párrafo anterior es solo un caso de las innumerables combinaciones que pueden resultar de la interacción normal con el sistema. Por eso, es necesario definir los siguientes tipos de interacciones de control:

- *Comienzo*: Indica el comienzo de una interacción que agrupa a más de una llamada a una función. Es similar al concepto de bloque de código en cualquier compilador y con esto, el submódulo de lenguaje natural se encargará de generar la construcción lingüística equivalente a la función que pasa a través de la interface, pero en lugar de enviarla directamente hacia el usuario del sistema (en el caso más directo sería la salida por otra interface fuera del módulo del tutor), la mantiene para agregarle todas las construcciones lingüísticas que se le indiquen a continuación, hasta recibir una interacción de control que le indique que puede despachar todos los "*mensajes*" que tiene en la cola de espera como si en realidad se tratase de un único mensaje, pero mas extenso.
- *Finalización*: Indica el final del bloque de interacción y avisa al submódulo de lenguaje natural que ya puede enviar todos los mensajes que mantiene encolados hacia su interface en la forma procesada de párrafo (o la construcción lingüística equivalente). Se podría agregar que no es necesario indicarle al submódulo de lenguaje natural que encole los mensajes que deben ser entregados a su interface de forma individual (como lo son cualquier `I_Response`, `I_Hint`, `I_Information`, `I_Question`) y por lo tanto no es necesario comenzar un bloque lingüístico

con `I_Control ("comienzo")`, asimismo, si esto se realiza, no se debería obtener un resultado diferente.

Luego, no se requieren más interacciones de control sobre la interface, pero el submódulo de lenguaje natural requiere mantener, como ya he explicado antes, el contexto sobre el que espera la respuesta, y eso lo obtiene, básicamente a partir de la interacción de respuesta del alumno ó puede obtenerlo también a partir de las interacciones de tipo pistas.

En el caso de que existan varias interacciones que requieran mantener contexto, la complejidad del módulo puede forzar a guardar sólo la última interacción de preguntar ó puede guardar todas y tomar la respuesta que mejor se adecue a alguno de los contextos que mantiene almacenados. Una vez finalizada la interacción inmediata puede eliminar el contexto (ó los contextos) almacenado, ya que la nueva interacción le corresponde al sistema. Otra opción sería que sólo borrara los contextos de las preguntas al recibir nuevas preguntas ó en el momento de recibir nuevas instrucciones de control, de esta forma se pueden aceptar varias respuestas individuales del alumno (aunque se viole el principio de que el alumno no comience ninguna interacción en el sistema, sino que simplemente continúe las interacciones que comienza el STI en una sesión pedagógica, aunque si se incluye este caso quedaría definida una interface mucho más flexible).

Pregunta real	Primitiva de Interacción
Un conjunto de elementos del tipo Ordinal es aquel que el compilador puede identificar el predecesor de un elemento iesimo del conjunto (menos el primero) y un sucesor del elemento iesimo (menos del ultimo).	I control ("Comienzo") I Information ("Conjunto ordinal") I Question ("concepto", "reales", "conjunto ordinal") I_control ("finalización")

– *Interacciones de tipo Mixtas:*

```
if  (input type == I control )  {
next_input()
         while (input type != I_control) {
switch (input) {
case I Information: next input()
encode text NL(Input) break;
case I Response: next input()
encode_text_NL(Input) break;
case I_Question: push (context)
encode_text_NL(Input) next_input() break;
case I_Hint: push (context)
encode text NL(Input) next input() break;
case I Control: break;
}
Output(processed_text)
Else {
switch (input) {
case I Information: next input()
encode text NL(Input) break;
case I Response: next input()
encode_text_NL(Input) break;
case I_Question: push (context)
encode_text_NL(Input) next_input() break;
case I Hint: push (context)
encode text NL(Input) next input() break;
case I Control: break;
}
Output(processed_text)
}
```

```
       ready=false;
       next_input()
       while (ready)
{ if stack (empty) == false) push (context)
else ready=true
concepts = decode_text_NL(Input,context)

       Evaluate concepts (concepts)
       Output(concepts)
```

Código Fuente 2: Ejemplo de procesamiento de entradas.

Este tipo de interacciones, son planteadas por Yujian [17] así como también por Hume [7], en el proyecto *CircSim* y es el caso donde el STI responde de manera compleja por medio de lenguaje natural. En el caso del esquema planteado, las interacciones del tipo mixto serían simplemente combinaciones del resto de las interacciones disponibles para la interface, pudiendo realizar cualquier combinación que salga del planificador de lección y de una longitud teórica sin limite (probablemente estará acotada en la práctica por la memoria disponible para la cola de mensajes de la interface del submódulo de lenguaje natural). A continuación Se presentan dos secciones de pseudocódigos aptos para trabajar con el modelo mencionado anteriormente para mantener los estados (ver Código Fuente 1 y Código Fuente 2). Estos pseudocódigos son válidos para el proceso que toma los datos de la interface y son procesados dentro del submódulo del lenguaje natural. Con esto no se está modificando el análisis realizado a este módulo visto como una "*caja negra*", sino que este pseudocódigo es para mostrar la ventaja de almacenar el contexto en el que se espera la respuesta y cómo está se procesa.

En el pseudocódigo planteado se puede ver que cada vez que se procesa una entrada (`next_input()`) el sistema identifica el tipo de esa entrada y si es una entrada de control, entra en el ciclo que le permite procesar todas las interacciones del bloque, de otro modo, solo procede a procesar la interacción individual. Cuando recibe una interacción de tipo pregunta (`I_Question`) ó de tipo pista (`I_Hint`), el submódulo almacena él (ó los) estados de la pregunta para facilitar la decodificación de la respuesta (`push (context)`).
Luego, realiza el proceso de construcción lingüística en el lenguaje natural (`Output(processed_text)`) y es el caso de una interacción mixta. La salida del sistema (`encote_text_N(Input)`) sólo se realizará cuando todas las interacciones individuales hayan sido procesadas, de otro modo, la salida será inmediata. En el caso de la decodificación de una respuesta, que utilice el contexto como ayuda, éste quedará almacenado.
Como se observa, se procesan las entradas del usuario con cada uno de los contextos disponibles (`decode_text_NL(Input,context)`) para obtener los conceptos de la respuesta, y una vez terminado de almacenar en `concepts` se procede a evaluar cuál de todas las transcripciones es la más adecuada para alguno de todos los conceptos (`Evaluate_concepts (concepts)`)

6. CONCLUSIONES Y TRABAJOS FUTUROS

Se presentaron los diferentes tipos de interacciones, su análisis a través de ejemplos y la forma de procesar cada una de ellas. Luego, se propone: a) Ampliar los contenidos disponibles para la autoevaluación, b) Escalar el sistema informático de tal forma que permita realizar un seguimiento del alumno, de esta forma el docente puede tener una clusterización de su clase en función de las necesidades cognitivas, c) Escalar el sistema informático hacia bases de datos e interfaces capaces de interactuar con el alumno de manera autónoma y d) Incluir un módulo de autoevaluación en los Sistemas Tutores Inteligentes cuya arquitectura se está desarrollando.

7. REFERENCIAS

[1] Cataldi, Z.; Salgueiro, F.; Lage, F. y García–Martínez, R. 2005. *Sistemas Tutores Inteligentes: redes neuronales para selección del protocolo pedagógico.* V WTIAE. XI CACIC: Congreso Argentino de Ciencias de la Computación. FCAD-UNER. Concordia. Entre Ríos. 17 al 21 de octubre.

[2] Di Eugenio, B.; Michael J. T. (2001*). Can simple Natural Language Generation improve Intelligent Tutoring Systems?.* Electrical Engineering and Computer Science Department. University of Illinois at Chicago.

[3] Evens, M. W.; Spitkovsky, J.; Boyle, P.; Michael, J.; Rovick, A. A. (1993). Synthesizing tutorial Dialogues. Preoceedings of the 15th Annual Conference of the Cognitive Science Society.

[4] Freedman, Reva. 2000. *Plan-Based Dialogue Management in a Physics Tutor. Proceedings of the Sixth Applied Natural Language Processing Conference* (ANLP 2000), Seattle.

[5] Graesser, A., Person, N., Harter, D., y TRG (2001). Teaching *tactics and dialog in AutoTutor.* International Journal of Artificial Intelligence in Education.

[6] Freedman, Reva. 1999. Atlas: A Plan Manager for Mixed-Initiative, Multimodal Dialogue. AAAI-99 Workshop on Mixed-Initiative Intelligence, Orlando.

[7] Hume G., Michael, J; Rovick, A.; Evens, M. (1996), Hinting as a tactic in one-on-one tutoring. Journal of Learning Sciencies.

[8] Evens, M. W.; Stefan, B.; Ru-Charn, C.; Freedman; Glass, M; Hee Lee, Y.; Leem Seop, Shim; Woo Woo, C.; Yuemei, Z.; Yujian, Z,; Joel, A. M.; Rovick, A. A. (2001). *CIRCSIM-Tutor: An Intelligent Tutoring System Using Natural Language Dialogue.* Twelfth Midwest AI and Cognitive Science Conference, MAICS 2001, Oxford, OH, p. 16-23.

[9] Reiter, E. y Dale, R. (2000) Building natural language generation systems Studies in Natural Language Processing Cambridge University Press.

[10] Huang, X. y Fiedler, A.. 1996. Paraphrasing and aggregating argumentative text using text structure. In *Proceedings of the 8th International Workshopon Natural Language Generation*, pages 21–30.

[11] Scott, D. and Sieckenius de Souza, C. .(1990). Getting the message across in rst-based text generation. In R. Dale, C. Mellish, and M. Zock, editors, Current Research in Natural Language Generation, pages 47{73. Academic Press.

[12] Urretavizcaya, M (2001) Sistemas inteligentes en el ámbito de la educación. Revista interamericana de Inteligencia Artificial N. 12. Pp. 5-12

[13] Sancho, L (2002) Una alternativa para el uso de computadoras en educación. Educación Net. Red global de educación a distancia (DistEdNET). Universidad estatal a Distancia.

[14] VanLehn, K (1988). *Student Modelling.* M. Polson. Foundations of Int. Tutoring systems. Hillsdale. N.J. Lawrence Erlbaum Associates, 55-78

[15] Wolf, B. (1984). Context Dependent Planning in a Machine Tutor. Ph.D. Dissertation, University of Massachusetts, Amherst, Massachusetts.

[16] Giraffa, L. M. M. (1997). Seleçao e adoçao de estrateguas de ensino em Sistemas Tutores Inteligentes . Porto Alegre: CPGCC/UFRGS.

[17] Yujian Zhoul, Freedman, R, Glass, M., Michael, J.; Rovick, A. y Evens, M. (1999).What Should the Tutor Do When the Student Cannot Answer a Question?. Proceedings of the Twelfth Florida Artificial Intelligence Symposium.

[18] Glass, M. (1997). Some Phenomena Handled by the CIRCSIM-Tutor Version3 Input Understander. Proceedings of the Tenth Florida Artificial Intelligence Research Symposium, Daytona Beach, FL, 1997, pp. 21–25.

Um Modelo para Seleção de Avaliações Adaptativas Em Ambientes Computacionais de Aprendizagem

Arthur S. Alves, Bruno W. R. Oliveira, Danilo M. Ikebara, Patrícia A. Bottaro, Renato Lopes, Edson Pinheiro Pimentel.

Universidade IMES
Av. Goiás, 3400, São Caetano do Sul, SP, Brasil, 09550-051

arthur.sapata@uol.com, bruno.willer@gmail.com, midanilo@gmail.com, abpatricia2002@yahoo.com.br, renatolps@superig.com.br, edson.pimentel@imes.edu.br

Resumo

No sistema tradicional de ensino e também na grande maioria dos Ambientes Computacionais de Aprendizagem todos os estudantes são avaliados de maneira uniforme independente do seu nível de aquisição de conhecimentos e dos conteúdos abordados. O insucesso recorrente nessas avaliações pode ser desestimulante para o aprendiz e torna o processo de avaliação formativa ineficaz uma vez que os resultados não são utilizados para realimentar o próprio processo de avaliação. Este artigo tem por objetivo apresentar um modelo para a seleção de avaliações adaptativas num ambiente computacional de aprendizagem utilizando técnicas de mineração de dados com base no nível de aquisição de conhecimentos do estudante em cada item do domínio em questão e também nos conteúdos abordados nas unidades de Avaliação. A seleção de unidades de avaliação adequadas ao perfil atual do estudante criará condições para avaliações personalizadas de modo a proteger ou desafiar o aprendiz nos seus sucessos ou insucessos.

Palavras chaves: Avaliações Adaptativas, Mineração de Dados, Avaliação Formativa.

Abstract

In the traditional system of education and also in the great majority of Computational Environments of Learning all the students are evaluated in an independent uniform way it its level of acquisition of knowledge and them boarded contents. The recurrent failure in these evaluations can be discouraged for the apprentice and becomes the process of inefficacious formative evaluation once that the results are not used to feedback the proper process of evaluation. This article has for objective to present a model for the selection of adaptive evaluations in a computational environment of learning using data mining techniques based on the level of acquisition of knowledge of the student in each item of the domain in question and also in the boarded contents in the units of the Evaluation. The selection of adequate units of evaluation to the current profile of the student will create conditions for personalized evaluations in order to protect or to defy the apprentice in its successes or failures.

Keywords: Adaptive Evaluations, data mining, Formative Evaluation

1. INTRODUÇÃO

O ensino com o uso da Tecnologia da Informação é uma modalidade que cresce a cada dia e com muita rapidez, seja ela no Ensino à Distância (EAD), no sistema híbrido que mescla o ensino presencial e o EAD, ou apenas nas atividades complementares ao processo de aprendizagem presencial. Em qualquer uma dessas categorias muitos ainda são os desafios e lacunas no que diz respeito ao desenvolvimento de sistemas computacionais que dêem suporte à Educação.

O processo de ensino-aprendizagem envolve três atores principais: estudante, professor e conteúdo. Integrando esses atores estão os métodos utilizados e as diferentes tecnologias que propiciam que o processo se complete. O mecanismo de aferição do processo nas mais diferentes etapas é a avaliação podendo ser diagnóstica somativa ou formativa [2].

No sistema tradicional de ensino e também na grande maioria dos Ambientes Computacionais de Aprendizagem todos os estudantes são avaliados de maneira uniforme independente do seu nível de aquisição de conhecimentos e dos conteúdos abordados. O insucesso recorrente nessas avaliações pode ser desestimulante para o aprendiz e torna o processo de avaliação formativa ineficaz uma vez que os resultados não são utilizados para realimentar o próprio processo de avaliação.

O ramo da inteligência artificial na área da computação dispõe de diversas técnicas que permitem o desenvolvimento de ambientes computacionais de aprendizagem dinâmicos e adaptativos. Sistemas Adaptativos devem ser capazes de se adequar ao aprendiz, no que diz respeito ao seu histórico e o seu desempenho [10]. Num contexto de avaliação formativa seria possível, por exemplo, exibir avaliações adaptadas ao perfil cognitivo do estudante, de tal modo que este aprendiz seria avaliado formativamente de acordo com os seus conhecimentos e suas lacunas de aprendizagem.

Este artigo tem por objetivo apresentar um modelo para a seleção de avaliações adaptativas num ambiente computacional de aprendizagem utilizando técnicas de mineração de dados com base no nível de aquisição de conhecimentos do estudante em cada item do domínio em questão e também nos conteúdos abordados nas unidades de Avaliação. A seleção de unidades de avaliação adequadas ao perfil atual do estudante criará condições para avaliações personalizadas de modo a proteger ou desafiar o aprendiz nos seus sucessos ou insucessos.

A seguir, a disposição das seções do artigo: a seção 2 apresenta uma revisão sobre conceitos de avaliação da aprendizagem, os problemas envolvidos na avaliação e a evolução do conhecimento medido através delas; a seção 3 descreve o ambiente NETEDU e suas ferramentas; a seção 4 apresenta um modelo para realizar a adaptação das avaliações e a seção 5 apresenta as considerações finais e trabalhos futuros.

2. AVALIAÇÕES

A Avaliação deve ter bem claramente definido o seu Objetivo. Com o objetivo estabelecido, certamente o nível de aprendizado desejado será alcançado. Segundo CHINEN[1], é a avaliação que determina a "evolução" do aluno, por isso a necessidade da determinação do real objetivo da Avaliação.

2.1 Classificações de Avaliação

A avaliação pode ser classificada, brevemente, em:

- Diagnóstica: ocorre durante o processo de aprendizado e consegue identificar os pontos fortes e fracos do aluno em referência ao conteúdo passado.

- Formativa: segundo PERRENOUD[2], ajuda o aluno a aprender a se desenvolver. A cada objetivo finalizado, indica ao professor e ao aluno o que foi realmente aprendido, evitando acúmulos de problemas, segundo PIMENTEL[3].

- Somativa: através de provas subjetivas, objetivas e classificatórias, são atribuídos notas e um feedback aos alunos quanto ao que foi aprendido ou não.

2.2 Propósitos da Avaliação

A avaliação, em sua totalidade, é um meio de mensurar ganhos. Mas o que fazer com essa informação, tem alguma funcionalidade?

Segundo SOMONSON [4], a resposta é sim. De fato, a adição de muitos usos que podem direta ou indiretamente influenciar o ambiente de aprendizado ou ajudar a formular políticas, existe muitas maneiras na qual a avaliação pode também melhorar o aprendizado. No ambiente de educação à distância, resultados de avaliações podem, algumas vezes, ser usados para comparar o desempenho acadêmica dos estudantes remotos com a performance daqueles que estão no local de origem da aula. Mas, para o estudante, a idéia de Avaliação é obter uma nota pelo seu desempenho, ao final do curso, ou mesmo ao final do conteúdo exibido pelo professor. No entanto, as notas provêm uma informação limitada, mesmo que elas sejam úteis para determinar o quanto pode ser melhorado a instrução de futuros estudantes, ou mesmo, do aluno avaliado.

O melhor propósito mais influenciável para avaliação do ganho de aprendizado por parte dos aprendizes é prover o feedback. Com isso os aprendizes ganham senso de controle e responsabilidade sobre o seu aprendizado.

2.3.Problemas da Avaliação

As avaliações que conhecemos hoje e que desde tempos remotos são aplicadas aos aprendizes da forma para todos. Não há particularidades. Os aprendizes são tratados do mesmo modo, ignorando as suas reais necessidades e seus conhecimentos.

Esse problema da Avaliação seria resolvido se o professor desse um acompanhamento individualizado aos aprendizes, tanto em questões instrucionais, quanto nas avaliações. Mas, e como realizar isso quando a turma de aprendizes é consideravelmente grande?

Um outro ponto muito importante na aprendizagem dos aprendizes é saber o que ele já sabia antes da instrução/aula dada pelo professor, e o que ele “acha” que sabe. Dois pontos extremamente

importantes para o bom andamento do processo de aprendizagem, e que nas Avaliações de hoje, pouco ou nada têm sido utilizadas.

2.4 Avaliações Adaptativas

A solução proposta por este artigo para a resolução dos problemas na avaliação dos dias atuais são as Avaliações Adaptativas. Segundo BRUSILOVSKI [11]-[12]-[13], a adaptatividade tem por objetivo adaptar o conteúdo de um nó de informação do domínio da aplicação ao nível decorrente de conhecimento e outras características do usuário.

As avaliações adaptativas que serão implementadas por meio das técnicas de adaptatividade, terão como objetivo dirigir determinadas questões da Avaliação para um determinado aprendiz, baseando-se pelo seu perfil.

Para resolver o problema da quantidade de alunos que se aplicaria a avaliação, FRANÇA [5] propõe que sejam formados grupos homogêneos compostos por alunos que possuam, mais ou menos, as mesmas lacunas de aprendizagem. No entanto, as questões adaptadas serão direcionadas ao Grupo de aprendizes e não mais ao aprendiz.

Para CURA [14], a base para a elaboração da avaliação adaptativa é o histórico de desempenho do aluno, que ficará armazenado e atualizado a cada avaliação aplicada. E a partir desse histórico, será possível, após as avaliações, gerar gráficos e estatísticas de desempenho dos conteúdos da avaliação e ter uma idéia melhor do acompanhamento da Disciplina por parte dos aprendizes.

3. O AMBIENTE NETEDU

Segundo CHINEN[1], o objetivo do ambiente é utilizar de sua arquitetura para criar condições de avaliar com mais freqüência a evolução do conhecimento do aprendiz. Além disso, o ambiente possui uma avaliação metacognitiva, onde o aprendiz irá refletir sobre os seus próprios conhecimentos.

3.1 A Avaliação no NETEDU

O ambiente, após a realização da primeira avaliação do aprendiz, gera as métricas metacognitivas KMA e KMB, e a métrica cognitiva NAC. Essas métricas ficarão “guardadas” no histórico do aprendiz, podendo, futuramente, criar gráficos de desempenho por conteúdo da disciplina (por aluno ou por avaliação).

Dando um destaque especial ao NAC (Nível de Aquisição de conhecimento), PIMENTEL [3] define como uma medida que índica o grau de conhecimentos do aprendiz em um determinado conteúdo, de um domínio de conhecimento, naquele instante.

Por meio do conhecimento dessas métricas, o professor é capaz de criar e direcionar as avaliações aos alunos, de acordo com o nível de conhecimento e as lacunas de aprendizado dos alunos, classificando os alunos em níveis. Porém, esse direcionamento é realizado de forma manual: o professor cria a avaliação para aquele determinado aluno (ou alunos), de acordo com o nível atual.

3.2 Classificação de aprendizes no NETEDU

Os aprendizes e as UA's (Unidades de Avaliação) no NETEDU possuem cinco níveis: Iniciante, Básico, Intermediário, Avançado e Experiente.

Segundo CHINEN [1]:

> *"Para conseguir a promoção de nível, é necessário obter sucesso em dois critérios pré-definidos pelo professor. O primeiro critério é a obtenção de uma "medida mínima" nos conteúdos da disciplina, configurada pelo professor. O segundo é a quantidade mínima de UA's respondidas relacionadas ao conteúdo da disciplina. Ao obter sucesso nos dois critérios, o ambiente vai automaticamente promover o aluno de nível, adaptando-o a um novo conjunto de avaliações, de acordo com o seu NAC, KMA e KMB."*

O primeiro critério faz referência ao que PIMENTEL [3] descreve como índice de conhecimento do NAC e o segundo critério refere-se ao índice de avaliação do NAC. Portanto, sendo o NAC sua principal métrica de avaliação, ela será usada como entrada de dados para o agrupamento de dados e a identificação da Unidade de Avaliação adequada.

4. UM MODELO PARA AVALIAÇÕES ADAPTATIVAS

De acordo com Brusilovsky et. al. [11], por hipermídia adaptativa entende-se todos os sistemas de texto ou hipermídia que refletem algumas funcionalidades do usuário e aplicam esse modelo para adaptar a vários aspectos visíveis do sistema para o próprio. Portanto, adaptatividade é usada para moldar a visão de um usuário em relação a um determinado conteúdo a seus requisitos pessoais. Tecnologias de hipermídia são muitas vezes utilizadas para guiar um usuário através de um material digital, ajudando na compreensão.

Sistemas de adaptatividade de hipermídia constroem um modelo de objetivos, preferências e conhecimentos de usuário individual, utilizando esse modelo através da interação para adaptar as necessidades do usuário.

Avaliações adaptativas são capazes de fornecer rotas eficientes e personalizadas no sentido de estabelecer o grau de conhecimentos de uma pessoa num domínio de conhecimento [9].

Nesse trabalho o que se pretende é criar um ambiente de avaliação capaz de indicar ao estudante unidades de avaliação que estejam de acordo com o seu nível de aquisição de conhecimentos.

Não é produtivo permitir que o estudante seja avaliado de maneira recorrente nos conceitos em que demonstra lacunas. Como existe uma hierarquia de conceitos especificando a dependência de pré-requisitos, torna-se possível mapear para o estudante avaliações mais adequadas ao seu perfil, de forma que este possa ser avaliado num crescente, até que este possa chegar em avaliações que englobem todos os conceitos do domínio em questão.

A figura 1 apresenta um esquema geral para o funcionamento da proposta de avaliações adaptativas neste trabalho.

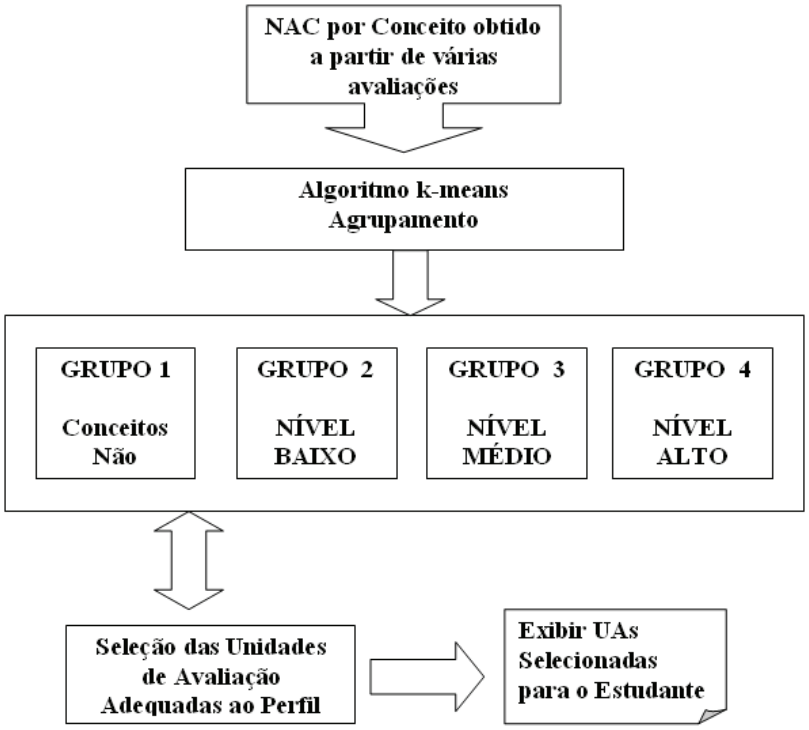


Figura 1 – Modelo para Adaptação das Unidades de Avaliação.

A seguir a especificação para a adaptação de avaliações de acordo com a proposta.

4.1. Agrupamento por Nível de Aquisição de Conhecimentos

Um dos problemas em Avaliações Adaptativas é se encontrar a melhor técnica para se adaptar as questões em um grupo de aprendizes com níveis de conhecimentos diferentes. Essas técnicas se baseiam em algoritmos que geram alguma saída que em muitos casos é de difícil compreensão.

Propõe-se neste artigo, num primeiro momento, a utilização de um algoritmo para agrupar os alunos e o respectivo NAC em cada conceito em quatro grupos: Conceitos não avaliados; Nível Baixo; Nível Médio e Nível Alto. Para esta etapa o algoritmo escolhido foi o K-Means descrito por Zuchini [8]..

O algoritmo K-means é considerado como um algoritmo de mineração de dados não supervisionado, serve para classificar ou agrupar seus objetos baseados em atributos/características no numero de K agrupamentos. O K-means recebe como entrada um número K de agrupamentos e atribui aleatoriamente um objeto como sendo o centróide inicial de cada agrupamento. Sucessivamente, cada objeto é associado ao agrupamento mais próximo e o centróide de cada agrupamento é então recalculado levando-se em conta o novo conjunto de objetos a ele pertencentes. O agrupamento é feito minimizando a soma dos quadrados das distancias entre dados e o centróide correspondente do conjunto. A seguir o funcionamento passo a passo do algoritmo:

a) Escolha K distintos valores para centros dos grupos (a escolha pode ser aleatória);

b) Associar cada ponto ao centro mais próximo;
c) Recalcular o centro de cada grupo;
d) Repetir os passos "b" e "c" até não haver alterações;

Um dos problemas apontados por Monteiro [6] e por Hoon [7] em relação ao algoritmo do k-means são que os clusters finais não representam uma otimização global, mas apenas local e clusters diferentes podem surgir a partir da diferença na escolha inicial aleatória dos centróides.
A Tabela 1 apresenta um exemplo de entrada para o K-Means. A primeira coluna identifica o estudante, a segunda coluna o conceito e a terceira coluna o NAC do estudante no respectivo conceito.

Tabela 1. Exemplo de entrada de dados para o K-Means

ALUNO	CONCEITO	NAC
101	A	5
101	B	3
101	C	6
201	A	3
201	B	8
201	C	7

A Tabela 2 apresenta um exemplo de saída do K-means com base na entrada especificada na Tabela 1. O algoritmo categorizou as linhas da Tabela 1 nas classes 1, 2 e 3.

Tabela 2. Exemplo de saída de dados do K-Means

CLASSE	ALUNO	CONTEÚDO	NAC
1	101	C	2
1	101	B	4
2	201	A	5
2	101	A	6
3	201	B	7
3	201	C	8

Para especificar qual das 3 classes é a classe baixa, média ou alta, calcula-se a média dos valores (NAC) na classe. Por exemplo, a classe 1 tem média 3 obtida pela somatória do NAC dividida pela quantidade de linhas da classe : (2+4)/2. A classe 2 tem média 5.5 e a classe 3 tem média 6.5. Basta então estabelecer um critério (regra de intervalo) para definir a categoria das classes.

O ambiente tomará por base as classes geradas para exibir as unidades de avaliação adequadas ao perfil do estudante conforme descrito a seguir.

4.2. Seleção das Unidades de Avaliação Adequadas ao Perfil

Com base na hierarquia de conceitos e espaços de conhecimento apresentados na seção 2 e nos grupos obtidos com o K-means as unidades de avaliação adequadas ao perfil do estudante serão selecionadas obedecendo aos seguintes passos:

a) O estudante deverá indicar em qual dos grupos pretende ser avaliado. Por exemplo, se desejar melhorar o seu NAC poderá escolher conceitos que estejam no grupo "baixo" ou "médio". Se desejar reforçar conceitos em que já possui nível bom desempenho selecionará o grupo "alto);

b) Se o objetivo do estudante for melhorar o seu NAC, o sistema selecionará dentro do grupo escolhido conceitos folhas ou conceitos que tenham o menor número de conceitos dependentes. Por exemplo, no grafo da figura 3, o conceito "1" é um conceito folha e o conceito "17" possui três conceitos dependentes, etc.

c) Selecionado o conceito, o ambiente deverá localizar UAs cadastradas que abordem o conceito, ou de forma isolada ou que o conceito possua um "peso forte". Os conceitos possuem "peso forte" em avaliações em que ele tem alto grau de participação no resultado final (nota final) da avaliação.

5. CONCLUSÃO

Um modelo para gerar avaliações adaptadas ao nível de aquisição de conhecimentos do aprendiz é apresentado neste trabalho. Para que o modelo possa funcionar adequadamente é necessário organizar o conhecimento na forma de pré-requisitos. Adotou-se a teoria dos espaços de conhecimento para tal. Para reduzir o espaço de busca no ato da adaptação pretende-se utilizar o algoritmo de agrupamento conhecido como K-means.

O Ambiente de Avaliação NetEdu ao qual o modelo adaptativo será incorporado foi apresentado mostrando que todo o seu arcabouço está preparado para receber no novo mecanismo que encontra-se em fase de implementação. Como aprofundamentos necessários pretende-se utilizar as métricas metacognitivas já existentes no NeTEdu no processo de gerar avaliações adaptativas.

Como trabalho futuro, pretende-se realizar estudos de casos reais para validar os ganhos obtidos com o uso de avaliações adaptativas.

REFERÊNCIAS

[1] CHINEN André Augusto Berti, SCHIRMER Felipe, CARRER Ricardo Nendre, SANTOS Rodrigo Henrique Costa dos Santos. Um ambiente para o Mapeamento do Conhecimento Cognitivo e Metacognitivo. Trabalho de Conclusão de Curso, Universidade IMES, 2006.

[2] PERRENOUD Philippe. Avaliação da excelência à regulação das aprendizagens – entre duas lógicas. Porto Alegre: Artmed,1999.

[3] PIMENTEL Edson Pinheiro. Um modelo para Avaliação e Acompanhamento Contínuo do Nível de Aquisição de Conhecimento do Aprendiz. Trabalho de Tese. Instituto Tecnológico de Aeronáutica, 2006

[4] SOMONSON Michael, SMALDINO Sharon, ALBRIGH Michael, ZVACEK Susan. Teaching and Learning at a Distance. Foundations of Distance Education, 3ª Edição.

[5] FRANÇA Vilma F. de , PIMENTEL Edson Pinheiro, OMAR Nizam. A identificação de grupos de aprendizes no ensino presencial utilizando técnicas de clusterização. Artigo apresentado no XIV Simpósio Brasileiro de Informática na Educação, 2003.

[6] MONTEIRO E. Silva, A. B., Portugal, M. S., Cechin, A. L. (2001). "Redes Neurais Artificiais e Análise de Sensibilidade: Uma Aplicação à Demanda de Importações Brasileira". Revecap vol. 5 n. 4.

[7] HOON, M. de; Imoto, S.; Miyano, S. (2002) "A Comparison of Clustering Techniques for Gene Expression Data" (abstract). Poster apresentado na 10th International Conference on Intelligent Systems for Molecular Biology em Agosto de 2002, Edmonton, Canadá.

[8] ZUCHINI, Márcio Henrique. Aplicações de Mapas Auto-Organizáveis em Mineração de dados e Recuperação de Informação. Trabalho de Mestrado apresentado a Faculdade de Engenharia Elétrica e de Computação (FEEC - UNICAMP), Universidade de Campinas, 2003.

[9] SITTHISAK, O., Gilbert, L. and Davis, H. C. (2007) Towards a competency model for adaptive assessment to support lifelong learning. In Proceedings of TENCompetence Workshop on Service Oriented Approaches and Lifelong Competence Development Infrastructures (in press), Manchester, UK.

[10] PIMENTEL, Edson Pinheiro, FRANÇA, Vilma Fernandes de, OMAR, Nizam. A caminho de um ambiente de avaliação e acompanhamento contínuo da aprendizagem em Programação de Computadores . Instituto Tecnológico da Aeronáutica, Universidade Municipal de São Caetano, Universidade Presbiteriana Mackenzie.

[11] BRUSILOVSKY, P. Methods ans Techinques of Adaptative Hypermedia User Modeling and User Adapted Interaction, 1996.

[12] BRUSILOVSKY, P. Adaptative Educational Systems on the World-Wide-Web. A Review of Available Technologies. In: Fifth Internacional Conference on Intelligente Tutoring System. ITS-98: San Antonio – Texas. 1998.

[13] BRUSILOVSKY, P. Adaptative Hypermedia. User Modeling and User Adapted Interaction. 2001.

[14] CURA Claudio, NUNES, Danilo, PIMENTEL, Edson P., BONANO, Enio, MANDAJI, Ricardo, OMAR, Nizam. Uma Ferramenta Adaptativa de Avaliação da Aprendizagem Baseada no Perfil Cognitivo e Metacognitivo do Estudante – Universidade Municipal de São Caetano do Sul, Universidade Presbiteriana Mackenzie.

Un modelo comprensivo para el diseño de procesos en una Aplicación E-Learning

Alejandro R. Sartorio

Departamento de Ciencias de La Computación, Universidad Nacional de Rosario (UNR)
Rosario (2000), Argentina
Centro Regional de Investigación y Desarrollo Rosario (CERIDER)
Rosario (2000), Argentina
sartorio@fceia.unr.edu.ar

Abstract

This paper describes an approach to documenting the conceptual design of E-Learning Web Transactions using UWATc+. A E-Learning Web Transaction is a collection of activities, which are formed by contracts context-aware (as software component), are that contributes to achieving a user-oriented pedagogical objective using a E-learning-based application. UWATc+ is a meta-model for describing the various aspects of a Web Transaction formed by contracts. It is an extension of the Transaction Design Model that is part of the Ubiquitous Web Applications (UWA) framework, a comprehensive framework for designing ubiquitous Web applications. A series of (extended) UML diagrams are used to graphically document the UWATc+ meta-model, which greatly facilitates adoption of the approach by practicing software engineers, expert in education, and programmers. Use of the approach for documenting Web Transaction is described through a case of study belonging to Obra Abierta project (CONICET - UNR).

Resumen

Este trabajo presenta UWATc+, una modelo conceptual y de diseño para la documentación de Transacciones Web E-Learning. Una Transacción Web E-Learning es una colección de actividades formadas por contratos con características context-aware (como componente de software), que permiten alcanzar un objetivo pedagógico orientado al usuario de Aplicaciones Web E-Learning. UWATc+ es un meta-modelo para la descripción de diferentes aspectos de una Transacción Web conformada con contratos. Esto es una extensión del Modelo de Diseño de Transacciones perteneciente al framework UWA (Ubiquitous Web Applications). Para documentar gráficamente el modelo UWATc+ fueron usados (extendido) diagramas UML, los cuales facilitaron su adopción por parte de los ingenieros en sistemas, expertos en educación y programadores. El uso de esta propuesta de documentación de Transacciones Web será descripto a través de un caso de uso perteneciente al proyecto Obra Abierta (CONICET-UNR)

1 INTRODUCCIÓN

A medida que el avance en la investigación y desarrollo de plataformas e-learning brinden mejoras e innovaciones de herramientas (videoconferencias, porfolios, wikis, workshops, etc.) y sus respectivos servicios, crece la cantidad de posibles configuraciones de los espacios e-learning. Abarcando diferentes tipos de requerimientos pertenecientes a las etapas de diseño, otros durante el desarrollo y otros en tiempo de ejecución. A partir de estos se definen los procesos e-learning (Pe-lrn). Al igual que los procesos de negocios en una Aplicación Web convencional (ej., www.ebay.com), los Pe-lrn están compuesto por transacciones Web [8]. En este contexto, una transacción (o transacción e-learning) es definida como una secuencias de actividades que el usuario ejecuta a través de una Aplicación E-learning con el propósito de efectuar una tarea o concretar un objetivo. El conjunto de actividades, sus propiedades y las reglas que gobiernan sus ejecuciones dependen del Pe-lrn que la Aplicación debe brindar.

Las características tecnológica de un Aplicación Web e-learning (AWe-lrn) son idénticas a las Aplicaciones E-learning convencionales (por ejemplo, las utilizadas en el proyecto de e-learning Sakai[1]) que proveen: navegación entre páginas a través de links, ejecución de Transacciones e-learning por medio de los servicios de las herramientas y las operaciones de un Pe-lrn. La principal diferencia tecnológica entre un AWe-lrn y una Aplicación e-learning convencional está en la incorporación de la teoría de coordinación de contratos [9, 1] en la implementación de algunos servicios que las herramientas brindan a los usuarios [2].

En base a la incorporación de los contratos en la implementación y diseño de Transacciones e-learning, aumenta las posibles configuraciones de los Pe-lrn. Comienzan a aparecer nuevas propiedades que tienen que ver con el campo de la Ingeniería de Software. Y, a su vez, fuertemente relacionados al trabajo multidisciplinario entre los actores de un proyecto de construcción de una AWe-lrn (ej. el proyecto Obra Abierta [2]). Para este fin, es imprescindible contar con un modelo de diseño que ayude en el ciclo de vida del desarrollo y configuración de una Aplicación e-learning. Integrando las tareas de los expertos en educación, diseñadores y desarrolladores. Desatender o resolver incorrectamente la documentación para el diseño de procesos tipo Pe-lrn pueden causar numerosos problemas reflejados en el proceso de configuración de un AWe-lrn. Estos problemas pueden ser: (1) Dificultades en la comunicación y entendimiento entre los clientes y diseñadores expertos en educación (primero), y entre los diseñadores y los desarrolladores (después), en el proceso de implementación de un AWe-lrn. (2) Determinación de las relaciones donde se justifique la inclusión de contratos. (3) Dificultad para visualizar la trazabilidad entre los procesos e-learning y las implementaciones de las Transacciones.

Este trabajo presenta un diseño compresivo para el modelado de los procesos de educación e-learning (Pe-lrn) en un Aplicación Web E-learning con la inclusión de los contrato con propiedades context-aware [2]. El modelo está basado en UWAT+ (Distante, 2004), una versión extendida y adaptada de "UWA Transaction Design Model" para el diseño de transacciones en aplicaciones Web. La principales contribuciones de este trabajo son: - El acercamiento de un modelo útil para la representación de transacciones e-learning en una AWe-lrn; permitiendo una mejor distinción de la ubicación (entre servicio-usuario(s) y servicio-

[1] SaKai: Entorno colaborativo y de aprendizaje para enseñar. Es de código abierto y está resuelto con tecnología Java. Url: wwww.sakaiprojet.org

[2] Obra Abierta: Proyecto de ID (CONICET), que se centra en el desarrollo e implementación de dispositivos hipermediales context-aware dinámico para investigar y aprender en contextos físicos-virtuales de educación superior. Directora: Patricia San Martín

servicio(s)) de la componente contrato dentro del flujo de ejecución. - Ejemplificar, mediante un caso de uso concreto, el uso del modelo UWATc+.

El resto de este trabajo se encuentra organizado de la siguiente manera: Tras esta introducción se presenta una referencia conceptual del los contratos, utilizados en el proyecto Obra Abierta (Sección 2). Luego se continúo con definiciones, formas de documentación y representación de Transacciones e-learning y procesos. (sección 3). Siguiendo, se presenta un modelo conceptual y de diseño sobre la adaptación propuesta para el modelado de Pe-lrn (sección 4). Mediante un caso de uso concreto, se describe el uso de UWATc+ para cada una de sus etapas. Finalizando con una breve conclusión.

2 CONTRATOS CONTEXT-AWARE PARA TRANSACCIONES E-LEARNING

En términos generales, la coordinación de contratos es una conexión establecida entre un grupo de objetos (en nuestras consideraciones, un objeto cliente y un determinado servicio). Cuando un objeto cliente (ej. un usuario alumno) efectúa una llamada a un objeto suministro (ej. un servicio de la herramienta Foro), el contrato "intercepta" la llamada y establece una nueva relación teniendo en cuenta el contexto del objeto cliente, el del objeto servidor, e información relevante adquirida y representada como contexto del entorno. Como condición necesaria, la implementación de los contratos no debe alterar el diseño y funcionalidad en la implementación de los objetos.

2.1 Elementos de la componente contrato

El contrato puede ser configurado por medio de diferentes mecanismos, desde el lenguaje cotidiano hasta un lenguaje de especificación formal y un lenguaje basado en XML para los casos que sean necesarias especificaciones que puedan ser procesadas por máquinas.

El tipo de tecnología y forma de implementación de los contratos es transparente para los objetos que consumen los servicios en donde se encuentran involucrados. La configuración de un elemento contrato que forma parte de las componentes de un servicio, representa la información necesaria del mismo para ser utilizado por el invocador, sin necesidad que el dicho objeto tenga detalles de la ejecución. El contrato representa una enriquecida y efectiva interface de construcción que contiene toda la información sobre las componentes de los servicios, y deberá tener referencias sobre algún tipo de información de contexto para su utilización.

En la figura 1 se puede observar los elementos conceptuales básicos de esta componente a través de una serie de elementos que lo relacionan.

Para una mejor comprensión de las componentes del modelo se explica su caracterización y funciones particulares:

Identificador – Una componente servicio es identificada para un determinado contexto por un único nombre en el espacio de nombre.

Comportamiento - De acuerdo con los roles asignados en un determinado contexto, una componente servicio expone comportamientos correspondientes a la provisión y pedido de operaciones, y/o publicaciones y recepción desde/hacia cada contexto. Las operaciones pueden ser definidas en dos tipos – operaciones que ejecutan cómputos o transformaciones (tipo "update") y operaciones que proveen algún tipo de información sobre consultas (tipo "query").

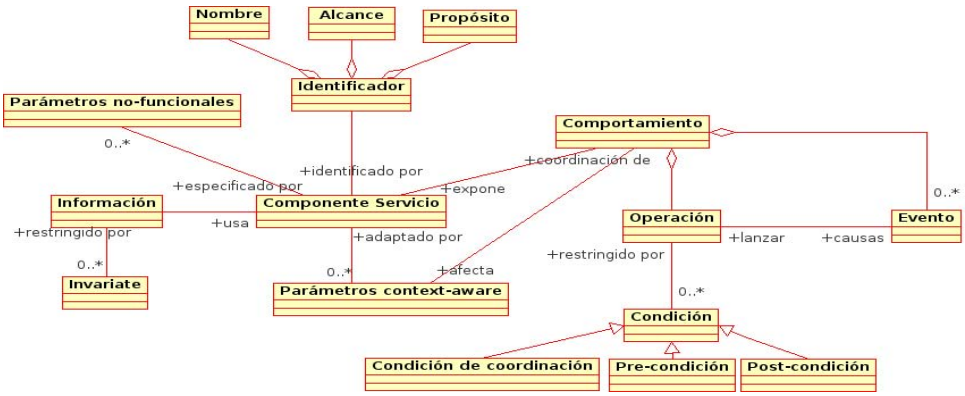


Fig. 1. *Meta-modelo del contrato context-aware*

Estas, se encuentran enteramente especificadas en base a un contrato, con el uso de pre-condición, post-condición y condicionales para lograr la coordinación entre contratos. En las condiciones de coordinación se especifican cómo requerir y proveer operaciones. Para lograr una comunicación precisa con una componente servicio, no sólo se tiene en cuenta qué operación fue provista o requerida y cómo el ejecutor ha lanzado el evento apropiado, sino también, cómo todas esas actividades están mutuamente relacionadas para ser aprovechadas por el objeto cliente. Un evento del contexto que lanza una operación dada, puede ser parte de un conjunto de pre-condición, mientras que un evento emitido a través de una exitosa operación puede ser parte de una pos-condición.

Tipos de Información – Una componente de servicio debe manejar, usar, crear o tener cierta información de recursos con el propósito de proveer servicios adecuadamente. Este elemento del contrato define el tipo de información relevante para las componentes asociadas al contrato, así como también restricciones y reglas sobre instancias de esos tipos. Esto representa un modelo de información lógica de una componente de servicio. Formalmente, esta información de tipos puede ser considerada como definiciones de tipos de los parámetros de las operaciones, o tipos relacionados a ellos.

Configuración de Parámetros Context-Aware – Una componente servicio depende del contexto de su actual entorno. Para utilizarse en diferentes contextos logrando la adaptación a eventuales cambios debe tener definido un conjunto de parámetros de configuración. Ejemplos de estos parámetros pueden ser: Contexto-del-Usuario (CU), locación en tiempo y espacio de los servicios consumidos y suministrados. Estos parámetros pueden ser enviados dentro de las invocaciones de las operaciones de los servicios o por medio de otros caminos, mediante componentes de servicios que pueden adaptar su comportamiento ante el cambio de contexto en una determinada situación.

Parámetros no funcionales – Una componente servicio puede definir un conjunto de los llamados parámetros no funcionales que caracterizan a la "calidad" de sus prestaciones dentro de un determinado contexto. Estos parámetros, son elementos para los consumidores de los servicios que permiten optar por el uso de un determinado servicio, o buscar otro con el mismo o similar contrato. Como ejemplo de parámetros no funcionales podemos mencionar: Performance, Fiabilidad, Tolerancia a Fallos, Costos, Prioridad y Seguridad.

3 DOCUMENTACIÓN DE LOS PROCESOS E-LEARNING

Las transacciones e-learning en un AWe-lrn están definidas como secuencias de actividades asociadas con un flujo de ejecución que permite al usuario desempeñar una determinada tarea y/o alcanzar una meta a través de la Aplicación. Entonces, un proceso e-learning (Pe-lrn) puede ser interpretada como una especificación del concepto de "workflow" en una Aplicación e-learning Web, con las condiciones (restricciones) que implica su concreción. En una Transacción e-learning Web, una *Actividad* está conformada por un conjunto de operaciones simples o complejas sobre datos y contenidos de la Aplicación. Como ejemplo de Transacciones e-learning se pueden mencionar un proceso en el cual un usuario (alumno) participa en un espacio de edición colaborativa (este caso será analizado posteriormente como caso de uso en las secciones posteriores)

Entonces, las transacciones en un AWe-lrn son el camino para la representación de los Pe-lrn, y proveer a los usuarios de servicios accederlos por medio de las herramientas que los contienen (wiki, foro, vídeo conferencia, taller, blog, etc.) La ejecución de Transacciones de un AWe-lrn supone tanto la navegación a través de las herramientas, por medios de los links de las componentes hipermediales, como el uso de sus servicios. Un ejemplo de servicio puede ser en una vídeo conferencia la posibilidad de que un docente edite en una pizarra compartida (con sus alumnos y colaboradores) una determinada ponencia.

El diseño de las transacciones abarca varios niveles de abstracción, distintos formalismos pueden ser usados para su representación y documentación. Al menos tres niveles de abstracción pueden ser representados: (1) nivel conceptual, (2) nivel lógico, y (3) nivel de implementación. El diseño conceptual permite una representación del sistema (las transacciones y su alcances) tal cual son percibidas" por el usuario y despejando las cuestiones de implementación. El diseño de la implementación se encuentra focalizado a proveer a los diseñadores de los AWe-lrn con todas las especificaciones necesarias para la configuración y realización de sus componentes El diseño lógico es un nivel intermedio del diseño de abstracción, utilizado para trasladar las especificaciones centrales del usuario desde el diseño conceptual hacia términos de especificaciones más cercana a las implementaciones.

Al igual que lo que ocurre con todos lo artefactos de software, el diseño de una Transacción e-learning para un AWe-lrn se puede tornar muy complejo. Cuanto más complejo sea el diseño, las confusiones entre los diseñadores (expertos en educación y analistas informáticos) y los implementadores (programadores y encargados de la Aplicación) crecerán. Para comunicar efectivamente la idea del diseñador es necesario una apropiada documentación. Si bien la documentación textual ha sido muy utilizada para describir detalles de implementaciones de bajo nivel, teniendo en cuenta que el diseño de las transacciones e-learning se describen en un nivel conceptual, es más adecuado una representación gráfica.

Existen diferentes aportes directamente relacionado a modelos "visuales" en forma de documentación gráfica [3–5]. En este contexto los modelos visuales son representaciones de sistemas de software que soportan múltiples perspectivas. Para el caso del diseño de las transacciones e-learning, una vista puede ser representada por una series de diagramas pertenecientes a UML (Unified Modeling Language) [12]

Los antecedentes relevantes que se relacionan con lo que entendemos por diseño de Pe-lrn, fueron estudiados de los aportes en el campo del métodos de diseños para Aplicaciones Web experimentados en los últimos años. Concretamente se pueden mencionar ADM (Atzeni y Parente, 2001), OO-H (Koch et al., 2003), OOHDM (Schmid and Rossi, 2004) y UWAT+ (Distante, 2005).

UWAT+ es un meta-modelo para la descripción de los distintos aspectos de Transacciones Web de manera holística. Es una extensión del Modelo de Diseño de Transacciones que forma parte del framework UWA (Ubiquitous Web Applications) [6]. Inspirado en este modelo y extendiéndolo para la contención de los contratos, se describe una adaptación para el diseño de Transacciones e-learning en un AWe-lrn.

4 UWATc+: una adaptación de UWAT+ para el modelado de procesos e-learning

Si bien los métodos de diseño de Transacciones de UWAT+ pueden ser utilizados para la representación de Transacciones e-learning, es necesario efectuar adecuaciones que tengan en cuenta la inclusión a los contratos (según sección 2). Tal cual fue mencionado en la sección 1, desde la perspectiva de los diseñadores, los contratos deben ser visto como una pieza de software para la instrumentación de los servicios de las herramientas. En consecuencia, es necesario tener un modelo que permita una mejor representación de los contratos, la visualización de su inserción en los servicios y las relaciones que en ellos representan (relaciones entre objetos que implementan servicios, usuarios y herramientas en la Aplicación).

La figura 2 se muestra un diagrama de clase UML que representa a los conceptos, las relaciones entre conceptos y los modelos para la representación de Transacciones. Los esquemas en color blanco pertenecen al modelo original UWAT+. El rectángulo y los esquemas grises describen los objetos, modelos y relaciones que que conforman el nuevo modelo denominado UWATc+.

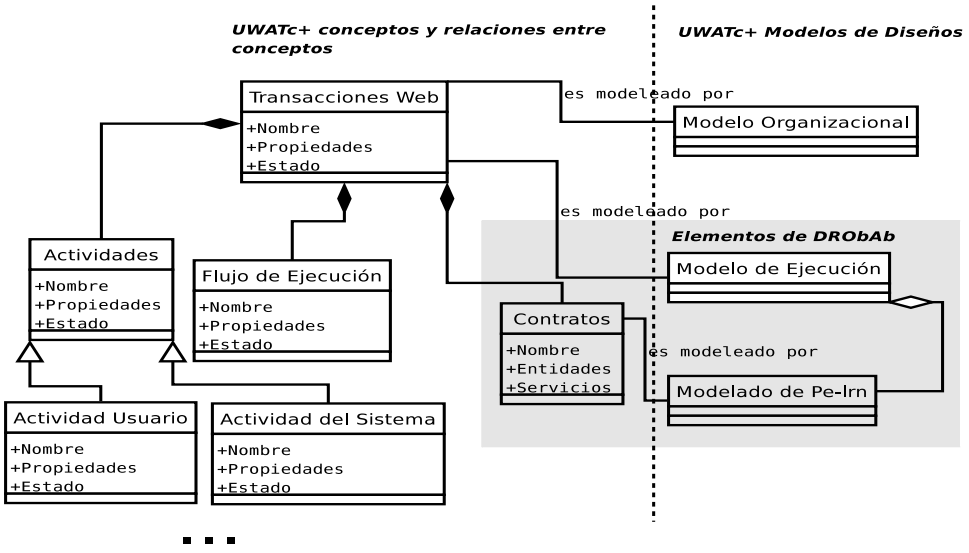


Fig. 2. *UWATc+ modelo conceptual y de diseño*

Como se describe en el diagrama, una Transacción Web es un objeto complejo (conceptual) compuestos por dos tipos de objetos principales pertenecientes al modelo original de UWAT+ y un tercer objeto agregado para la representación de los contratos pertenecientes a

las Transacciones e-learning. En el primer grupo se encuentra *Actividad* para la distinción de las actividades de los usuarios y del sistema. El objeto $FlujodeEjecución$ representa el orden lógico y temporal para la ejecución de las actividades comprendidas en las Transacciones. A su vez, una Transacción Web puede ser descripta por el $ModeloOrganizacional$ (desde el punto de vista estático) y el $ModelodeEjecución$ para la definición de las reglas de ejecución de la componente actividad (desde el punto de vista dinámico).

Cuando una Transacción Web contiene un contrato (definida como transacción e-learning) debe ser incluida una nueva componente para el diseño (representada en la figura como una relación de agregación en el $ModelodeEjecución$), conjuntamente con un nuevo modelo de diseño, $ModelodePe-lrn$, que permitirá la representación del contrato (caracterizada como relación de asociación con el objeto *Contrato*).

De esta manera quedan conformados los elementos que componen el modelo UWATc+ (rectángulo gris) y sus relaciones con el modelo original UWAT+. A continuación se describe en detalle el modelo usado por UWATc+ para el diseño de los transacciones que utilizan contratos (transacciones e-learning).

4.1 Requerimientos para el diseño de Transacciones e-learning

En esta sección se describen la caracterización de dos tipos representativo de requerimiento que motivaron la creación de este nuevo modelo de diseño (definidos en la sección 1). En base a experiencias recogidas por el grupo del proyecto Obra Abierta en el diseño y configuración de Aplicaciones e-learning, se presentarán dos tipos de requerimientos que deben ser cubiertos por el modelo de diseño. En primer lugar se enuncian cuestiones técnicas de diseño (desde el punto de vista de la Ingeniería de Software), seguido de un comentario sobre el tipo de Transacciones que el diseñador puede especificar a través del modelo.

- Especificar como son afectadas la ejecución de las actividades por los objetos contratos.
  La relación de una actividad con un contrato se produce cuando existen objetos interrelacionados por medio de un contrato [9]. De esta manera, el diseñador debe poder documentar las información de los objetos involucrados, sus métodos y parámetros. El contrato representa un tipo diferente de relación a la original entre los objetos, con la propiedad de reconfiguración en tiempo de ejecución. Característica que permiten una mejor adaptación a los requerimientos que resuelven las Transacciones e-learning.
- Definir cuál y cómo la información de los objetos contratos ("information object" [13]) es afectada por la ejecución de las Actividades.
  Una actividad funcional consiste en la ejecución de uno a más operaciones elementales (inserción, borrado, modificación, etc.) sobre los datos de la Aplicación y la información de los objetos contratos envuelta en la actividad. El modelo de diseño de Transacciones e-learning debe permitir al diseñador definir cuales operaciones del contrato son fundamentales para cada actividad, modelando el camino en que cada actividad elemental afecta la información de los objetos involucrados (modificando sus instancias por medios de sus ejecuciones).

5 DISEÑO DE PROCESOS E-LEARNING CON UWATc+

En esta sección se describen los resultados de la Aplicación de UWATc+ para el espacio dedicado al libro "Hacia un dispositivo hipermedial context-aware Dinámico. Educación e

investigación para el campo audiovisual interactivo" (San Martín P. 2007, et. al) [2]. Este modelo fue aplicado parcialmente para el diseño de los requerimientos fundamentales y el modelado del comportamiento funcional enmarcado bajo la perspectiva de proyecto Obra Abierta - mencionado anteriormente. Se presentará un caso de uso para ejemplificar el diseño de un Pe-lrn en la Aplicación e-learning para Obra Abierta. El diseño describe parte del proceso en que un usuario hace uso de los servicios de edición de la herramienta Foro a través de un contrato.

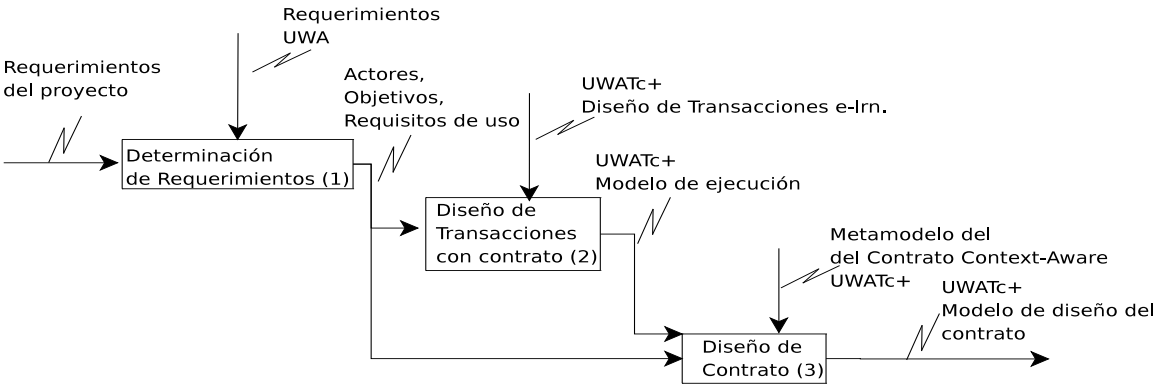


Fig. 3. *El proceso de diseñar Pe-lrn en un AWe-lrn con UWATc+*

El diagrama de la figura 3 muestra una adaptación del diseño de procesos de UWA [6] que se utiliza en UWATc+ para el diseño de transacciones con contratos. Para ilustrar el proceso de diseño se utiliza la notación IDEF0 (IDEF-0, 1993). En comparación con la metodología original de UWAT+, se agregaron las fases del diseño de contrato y fue modificado el modelo de diseño de la transacciones. Además, fueron excluidas las fases de "diseño de la información" y "diseño de publicación". Las fases del procesos de diseño son: (1) Determinación de Requerimientos; (2) Diseño de transacciones con contrato; (3) Diseño de contrato.

5.1 Determinación de Requerimientos

La fase de Determinación de Requerimientos toma como entrada las especificaciones del proyecto y produce, por medio de un mecanismo de refinamiento, las siguientes salidas: Cada actor con su objetivos relativo. Los requerimientos para la construcción y configuración de un AWe-lrn.

El modelo utilizado es orientado a objetivos: cada actor se identifica con al menos un objetivo, i.e, una abstracción de los objetivos que a través de la aplicación se debe alcanzar; cada objetivo es refinado en otros sub-objetivos, hasta poder definir el requerimiento en un bajo nivel suficiente para poder ser implementado. Este fase es similar a la descripta en (UWA Consortium, 2001) [6].

Para este caso de uso, atendiendo a los lineamientos en [2], se describe parte del modelo original donde se caracterizan los objetivos involucrado con un actor del sistema. A su vez, a medida que se van derivando los sub-objetivos comienzan a establecerse requerimientos concernientes a la teoría de coordinación de contratos context-aware [1, 9]. En la figura 4 dicha situación ocurre en la derivación de las tres ramas de objetivos y sub-objetivos, influyendo

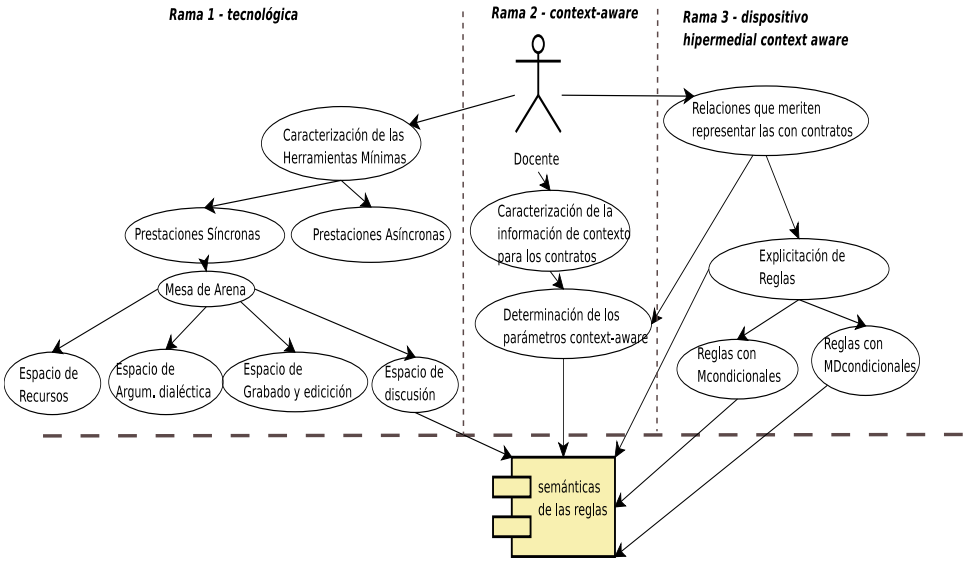


Fig. 4. *Objetivos de al alto nivel para un Pe-lrn*

directamente en la composición de la componente contrato. En este caso, tomando desde la ***rama 1*** un servicio de una herramienta de un espacio de discusión (herramienta Foro); de la ***rama 2*** se desprenden las información necesaria para poder articular dicho servicio teniendo en cuenta la información de contexto; la ***rama 3*** aporta el consenso de los expertos (del dominio e-learning) para la inclusión de los contratos en aquellas relaciones que mantendrán las propiedades de la Aplicación e-learning [1]. A través de este modelo, se logra un primer acercamiento sobre como se relaciona un contrato con: los requerimientos, el tipo de objetivos para cada requerimiento y los actores del sistema.

5.2 Diseño de Transacciones e-learning

El diseño de transacciones e-learning retoma la misma idea y modelo propuesto por UWAT+ para el diseño de transacciones [7]. Partiendo de los resultados de la fase de *Determinación de Requerimientos*, fundamentalmente de la caracterización de los contratos, pueden ser seleccionados una series de transacciones, i.e., objetivos que requieran la ejecución de una o más Actividades para su cumplido. Para cada uno de los objetivos que incluya contrato deben ser diseñadas Transacciones e-learning (de igual forma que con las transacciones en UWAT+), los cuales en principio deben ser establecidos desde un punto de vista estático (en este trabajo no abordaremos tal consideración) y luego, desde un punto de vista dinámico por medio de un modelo de ejecución. En la figura 5 se muestra una porción del modelo de ejecución de un transacción e-learning cuyo contrato asociado fue caracterizado en la fase de *Determinación de Requerimientos* (figura 4, sección 5.1). En el diseño se describe el flujo de ejecución entre las Actividades de la transacción. El modelo de ejecución es una adaptación del diagrama de actividades de UML [11] en el que las Actividades y sub-Actividades están representadas por estados (óvalos), y el flujo de ejecución entre ellos se representa por medio de transiciones (arcos). Los óvalos con el símbolo (*) - un asterisco entre paréntesis - refiere a una Actividad que representa a un conjunto de Actividades compuestas, y cuyo modelo de ejecución debe

ser representado con otro diagrama. Un óvalo simple representa una Actividad Elemental. Un óvalo color gris indica una Actividad compuesta de las sub-actividad que se encuentra dentro. Una sub-Actividad representada con un óvalo color gris indica que es dependiente de la Actividad que la contiene, esto quiere decir que su ejecución estará acompañada por otra sub-Actividad y no puede ser incluida en otra composición. Los arcos de lineas continuas indican flujos de ejecución obligatoria (transacciones hacia Actividades requeridas), mientras que los actos con lineas de puntos representan flujo de ejecuciones opcionales (transacciones hacia Actividades opcionales).

Cada relación posible entre actividades es representada por medio de una arco entre ellas. A cada arco se le asocia un texto que indica bajo que condiciones se produce la transacción, o el resultado de la ejecución de la Actividad de origen. Para describir como colaboran los usuarios de la aplicación en la ejecución de la Actividades puede ser anexado un diagrama UML Swimlanes.

Cuando una Actividad ejecuta servicios implementados por contratos, entonces, se establece un arco saliente hacia un contrato. El contrato tiene un nombre, y entre paréntesis se indican cuales son los objetos participantes (en el caso de tener ese tipo de información). Para representarlo visualmente se utiliza el estereotipo del elemento componente de UML. Las Actividades que influyan en la modificación de los contratos en tiempo de ejecución se conectan a través de un arco de linea de puntos, igual a los utilizado en la representación de los flujos opcionales. Los detalles de implementación del contratos se detallan en un diagrama aparte, perteneciente a la fase descripta en la siguiente sección.

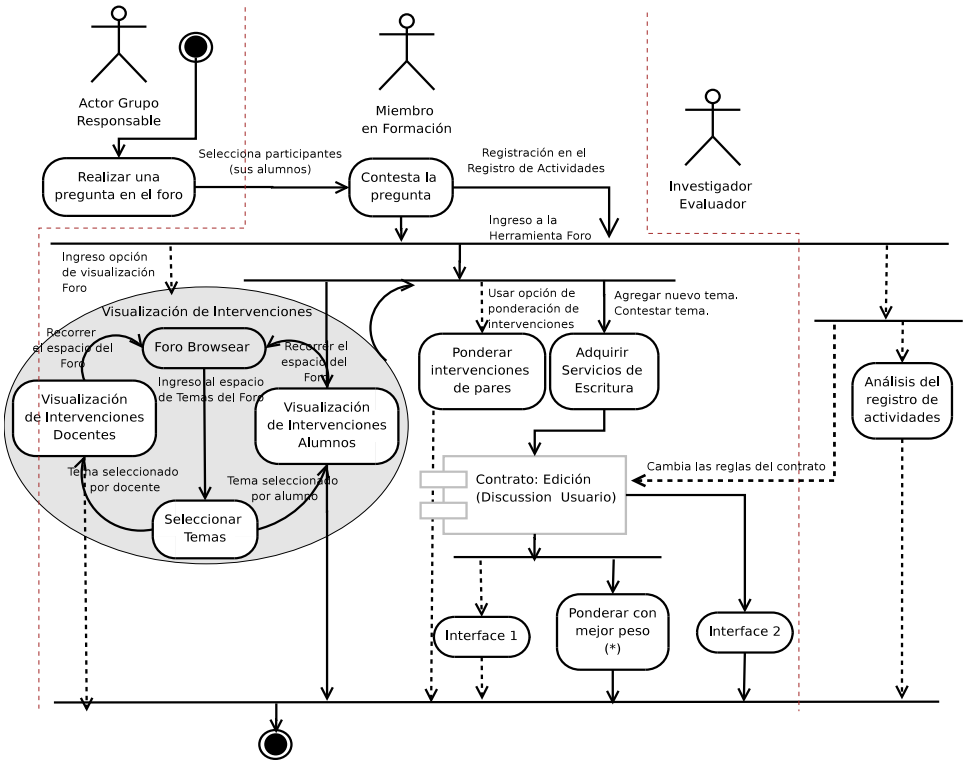


Fig. 5. *Modelo de ejecución de Procesos con Transacciones e-learning*

Por ejemplo, una de las opciones de la herramienta Foro de Obra Abierta ($http://200.80.157.171:8080/portal$) es la visualización de las intervenciones de los usuarios en el Foro. Un usuario docente puede seleccionar la opción "Foro" de la página principal de la Aplicación, luego seleccionar el tipo de "vista" (mediante un "comboBox") para ingresar en modo "browser" donde se muestran las intervenciones de los usuarios por temas. Una vez seleccionado el tema (por medio de la Actividad "Seleccionar Tema", figura 5), es posible ingresar al espacio de las intervenciones de los usuarios por medio de los roles de docente o alumno. Los docentes y alumnos tienen diferentes tipos de "vistas", permisos y servicios asociados (representadas en las actividades: "Visualización de Intervenciones Docentes" y "Visualización de Intervenciones Alumnos"). A través del Browser (Actividad "Foro Browser") se recorre todo el contendido del espacio y al mismo tiempo puede ser seleccionado otro tema para visualizar.

En cambio, si la opción seleccionado es añadir o responder temas, se ingresa a una página Web configurada para editar texto (por medio de la Actividad "Adquirir servicios de escritura"). Algunos de los servicios de edición y configuración de opciones son implementadas a través del contrato "Edición" (representado por la figura de la componente UML, con el contorno color gris). El flujo de ejecución, luego de la intervención de los contrato, dependerán de las reglas de coordinación y se representan con los arcos salientes similares a los usados para representar las relaciones entre estados.

5.3 Diagrama de un Contrato

Existen diferentes formas de representación de los contratos definidos en la sección 2, la herramienta CED (Coordination Development Environment) [3] los implementa a través de un lenguaje llamado Oblog [10]. En [14] se muestra como a través de CommUnity se definen primitivas de modelado y técnicas de diseño basadas en la separación de la "coordinación" del "cómputo".

En UWATc+ se brinda un diagrama de representación de contrato, donde se describen todos los datos que lo instancian. Cada tipo de dato y valor, pertenece a un elemento del meta-modelo de la figura 1. Teniendo en cuenta la figura 6, en primer lugar (item 1) se identifican los objetos participantes en el contrato; en el ejemplo de la figura 6 $DiscussionAction$ y $UserAction$ hacen referencias a dos clases reales perteneciente a la implementación de la herramienta Foro y Usuarios de la Aplicación Obra Abierta, respectivamente. Luego, se identifican los nombres de los parámetros context-aware significativos para el contrato, alineados en la misma columna del objeto que lo comparte (item 2). En Servicios (item 3) deben ser representados los métodos del objeto, que al ser ejecutados, provocan la intervención del contrato. Para este ejemplo $initState$ y $getIdentifier$ son ejecutados cuando un usuario ingresa a la herramienta Foro y las posteriores funcionalidades (servicios) disponibles dependen de la ejecución del contrato *Edición* (la figura 5 muestra la superposición del contrato entre los servicios de edición y las nuevas interfaces o funcionalidades). Las siguentes filas (items 4 y 5) se refieren a las pre y post-condiciones que se deben cumplir en la ejecución del contrato. Por último se explicitan las reglas de coordinación. Siguiendo con el ejemplo, en la parte del condicional $u.contexto =' l1; p1; docente; r1; c1;'$ verifica si el contexto del usuario u está compuesto por la locación $l1$, tienen el perfil $p1$, es un *docente*, cumple el rol $r1$ y pertenece a la categoría $c1$ (este tipo de representación de contexto se encuentra desarrollado en [2]). En

[3] CED: es el primer prototipo de una herramienta que implementa el uso de la coordinación de contrato en aplicaciones Java. La herramienta pertenece a ATX Sotfware (www.atxsoftware.com.ar);fue desarrollada en Java y es de código abierta.

cuanto a la acción de la regla de coordinación, continuando con el mismo ejemplo, se induce la ejecución del método *showMessage* del objeto *d* (DiscussionAction). El final del diagrama está dedicado a comentarios generales; cada comentario debe ir acompañado con el número de item (1,2,3,4,5 o 6) al que hace referencia.

Contrato: Edición			
1.	*Participantes*:	**d**:DiscussionAction	**u**:UserAction
2.	*Param. c-a*:	state, portlet, rundata, context	contextidentifier, identifier
3.	*Servicios*:	initState()	getIdentifier()
4.	*Pre-Cond*:	existe < *contexto* >	existe < *contexto* >
5.	*Pos-Cond*:	modifica < *contexto* >	
6.	*Reglas de Coordinación*:	**Si u**.contexto='p1;d;r1;c1;' **entonces d**.showMessage(data,string)	
Comentario			
1. DiscussionAction y UserAction pertenecen a clases implementadas en JAVA del proyecto Sakai. 4 y 5. < *contexto* > refiere a un objeto donde se oculta toda la información de contexto que caracteriza a los usuarios de la plataforma			

Fig. 6. *Diagrama del contrato: Edición*

6 CONCLUSIÓN

En base a la experiencia recogida en el proyecto Obra Abierta, es posible asegurar que la implementación del modelo de diseño UWATc+ en el ciclo de vida del desarrollo Aplicaciones E-Learning permitió un mejor entendimientos entre los expertos en educación, diseñadores y programadores. El modelo también ayudó en la compresión de la teoría de coordinación de contrato aplicada a transacciones para Aplicaciones e-learning.

References

1. Sartorio A., San Martín P. Sistemas Context-Aware en dispositivos hipermediales dinámicos para educación e investigación. *Universidad Nacional de Quilmes (UNQ). Capítulo. En impresión.*
2. San Martín P., Sartorio A., Guarnieri G., De la Riestra M. Hacia un dispositivo hipermedial context aware dinámico. Educación e Investigación para el campo audiovisual interactivo. *Universidad Nacional de Quilmes (UNQ). Libro. En impresión.*
3. Hartmann J., Huang S., and Tilley S. Documenting Software Systems with Views II: An Integrated Approach Based on XML. Proceedings of the 19th Annual International Conference on Systems Documentation (SIGDOC 2001: October 21-24, 2001; Santa Fe, NM), pp. 237-246. ACM Press: New York, NY, 2001.
4. Tilley S. and Huang S. Documenting Software Systems with Views III: Towards a Task-Oriented Classification of Program Visualization Techniques. Proceedings of the 20th Annual International Conference on Systems Documentation (SIGDOC 2002: October 20-23, 2002; Toronto, Canada), pp. 226-233. ACM Press: New York, NY, 2002.
5. Tilley S., Müller H. and Orgun M. Documenting Software Systems with Views. *Proceedings of the 10th Annual International Conference on Systems Documentation (SIGDOC '92: October 13-16, 1992; Ottawa, Canada), pp. 211-219. ACM Press: New York, NY, 1992.*
6. UWA Consortium. Ubiquitous web applications. *Proceedings of The eBusiness and eWork Conference (e2002), 16–18 October, Prague, Czech Republic.*
7. Distante D., Tilley S. and Huang S. (2004b). Documenting software systems with views IV: documenting web transaction design with UWAT+. P*roceedings of the 22nd International Conference on Design of Communication (SIGDOC 2004), Memphis, TN, New York, NY: ACM Press, 10–13 October.*
8. Brambilla M., Ceri S., Fraternali P. and Manolescu I.Process modeling in web applications. *ACM Transactions on Software Engineering and Methodology (TOSEM), in print.*
9. Andrade L. and Fiadeiro J.L. Interconnecting Objects via Contracts. I*n UML'99 – Beyond the Standard, R.France and B.Rumpe (eds), LNCS 1723, Springer Verlag 1999, 566-583.*
10. The Oblog Corporation. The Oblog Specification Language. http:// www.oblog.com/tech/spec.html
11. Object Management Group (OMG). Unified Language Modeling Specification (Version 2.0). *Online at www.omg.org, 2004.*
12. Murphy S., Tilley S. and Huang S. The 4th Workshop on Graphical Documentation: UML Style Guidelines. *To be held as part of The 22nd Annual International Conference on Design of Communication (SIGDOC 2004: October 10-13, 2004; Memphis, TN)*
13. Distante D., Tilley S. and Huang S. Documenting software systems with views IV: documenting web transaction design with UWAT+. *Proceedings of the 22nd International Conference on Design of Communication (SIGDOC 2004), Memphis, TN, New York, NY: ACM, 10–13 October.*
14. Gelernter D. and Carriero N. Coordination Languages and their Significance. *Communications ACM 35, 2, pp. 97- 107, 1992.*

Un Sistema Experto de Ayuda para el Ingreso de notas en la FCACE

Lisbeth I. Sante[1] **- Marco A. Coral** [2]
FISCT. Universidad Inca Garcilaso de la Vega [*]
Lima, Av. Bolívar 1848 – Lima 21, Perú
[1] lsantem@uigv.edu.pe, [2] mcoral@uigv.edu.pe

and

David Mauricio
FISI. Universidad Nacional Mayor de San Marcos [**]
Lima, Av. Germán Amézaga s/n, Lima 21, Perú
dms_research@yahoo.com

Abstract

The help systems are also know as support systems online and they are essential in the interaction processes between human being and computer, these provide brief information to resolve one problem during the user's job, besides they can get a high effectiveness in the use of computer applications at the productive or operational level, with these systems the user experiments that the product is intelligent and friendly, so that, is necessary use it for the computer applications of the Institutions. An alternative of help is the expert systems, which can improve the performance and the learning in the use of computer applications. This paper presents an expert system of help based on the method of progressive linking for using note's modules of Ciencias Administrativas y Ciencias Económicas Faculty, Inca Garcilaso de la Vega University, that the module takes a base of knowledge, a base of events and an inference engine with an algorithm defined such as: all the solutions – with priority.

Keywords: Help systems, computer applications, expert systems, progressive linking.

Resumen

Los sistemas de ayuda son conocidos también como sistemas de apoyo en línea y son esenciales en los procesos de interacción humano-computador, estos proporcionan información breve y concisa para resolver un problema puntual durante el trabajo del usuario, además pueden lograr una alta efectividad en el uso de aplicaciones informáticas a nivel productivo u operacional, con estos sistemas el usuario experimenta que el producto es inteligente y amigable por ello se hace necesario utilizarlos para las aplicaciones informáticas de las instituciones. Una alternativa de ayuda son los sistemas expertos los cuales pueden mejorar el rendimiento y el aprendizaje en la utilización de aplicaciones informáticas, el trabajo en mención presenta un Sistema Experto de ayuda basado en el método de encadenamiento progresivo para el uso del módulo de notas de la Facultad de Ciencias Administrativas y Ciencias Económicas de la Universidad Inca Garcilaso de la Vega, el cual maneja una base de conocimientos, una base de hechos y un motor de inferencia con un algoritmo definido como: todas las soluciones - con prioridad.

Palabras Claves: Sistemas de ayuda, Aplicaciones Informáticas, Sistemas Expertos, Encadenamiento progresivo.

[*] FISCT - Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones
[**] FISI – Facultad de Ingeniería de Sistemas e Informática

1. INTRODUCCIÓN

Un sistema experto puede definirse como un sistema informático que simula a los expertos humanos en un área de especialización dada [6], [8], [5]. Por esta razón el sistema debe ser capaz de procesar y memorizar información, aprender y razonar en situaciones definidas, tomar decisiones y explicar las razones de dichas decisiones [4]. En la actualidad los sistemas expertos presentan enormes ventajas en sistemas de ayuda a los usuarios en determinados campos donde no son expertos [2], en la FCACE* los usuarios que no manejan herramientas informáticas supera largamente el promedio, esta realidad genera retrasos y deficiencias en el uso de los sistemas automatizados para mejorar la gestión académica y administrativa de la facultad [15]. La mayoría de estos problemas recaen en el uso del módulo de notas con la cual se generan múltiples errores como ausencia de notas, equivocaciones, quejas continuas de los usuarios, reportes errados, etc.

En el presente trabajo se propone un sistema experto de ayuda para los usuarios al utilizar el modulo de notas académico para lo cual se requiere de una base de conocimientos [20], [22], [8] (serie de Reglas "Antecedente 1, Antecedente 2 → Consecuentes"), una base de hechos (proporcionado de acuerdo al caso) y un método o heurística [9] con el cual el motor de inferencia determine los resultados para cada caso [19], [5].

El trabajo hace una breve referencia inicial a los sistemas de ayuda y los sistemas expertos; la lógica desarrollada con el método de encadenamiento progresivo, la construcción de la base de conocimiento, la elaboración del algoritmo y su corrida respectiva, en la sección 4 el Software construido y finalmente las conclusiones.

2. LOS SISTEMAS DE AYUDA

Los sistemas de ayuda o soporte para el usuario pueden clasificarse en tres tipos: tutoriales en línea, documentación en línea y ayuda en línea; aunque en la práctica resulta muy poco para las necesidades del usuario, ya que en todo momento el manejo de cualquier aplicación necesita un soporte dinámico y contextual que ayude a su manejo, e integre todas las técnicas necesarias para solventar cualquier tipo de necesidad del usuario. Para construir un sistema de ayuda se deben tener en cuenta los objetivos de usuario, la tabla No.1 nos muestra una clasificación para sistemas de ayuda desde la perspectiva del usuario, se desprende para la utilización de una aplicación informática y se hace necesario un sistema de ayuda en línea que interactué con él..

Objetivo del usuario	**Medio de distribución**	
	Impreso	**En línea**
Comprar	- Folleto de venta, Hoja de compra	- Un software "demo"
Aprender	- Un Tutorial	- Un paseo (Tour) por el Programa
Usar	- Un manual de usuario	- Un documentación online

Tabla No. 1: Clasificación del material de ayuda [16].

* FCACE = Facultad de Ciencias Administrativas y Ciencias Económicas.

Los sistemas de ayuda para aplicaciones informáticas son programas que asisten al usuario en su utilización [12], [2], de ahí concluimos que su propósito principal es proporcionar respuestas a los problemas específicos que se presentan en el uso de dichas aplicaciones [7], estos sistemas incorporan técnicas de inteligencia artificial y pueden ser desarrollados utilizando sistemas expertos [1], [3].

2.1. Los sistemas expertos.

Los sistemas expertos proceden inicialmente de la inteligencia artificial la cual puede ser definida como un conjunto de cualidades informáticas que presentan características similares a la de la inteligencia humana lo cual permite resolver problemas a través de la percepción, el entendimiento, el aprendizaje y el razonamiento [17].

Los sistemas expertos están enfocados a un dominio específico donde se busca simular el razonamiento de un experto humano [2], estos son buenos para predecir resultados futuros a partir del conocimiento que tienen [20], [19], una característica importante de los Sistemas Expertos es la separación entre conocimiento (reglas, hechos) por un lado y su procesamiento por el otro. A ello se añade una interfaz de usuario y un componente explicativo de donde se desprende la arquitectura clásica [10], [18].

Estos sistemas están constituidos por una serie de elementos que unidos pueden encontrar respuestas a casos concretos formulados o sometidos a consideración; estos elementos son:

- Una base de Conocimiento.
- Una base de hechos.
- El Motor de Inferencia.
- Modulo de adquisición del conocimiento
- Modulo de Explicación.
- Interfaz de Usuario.

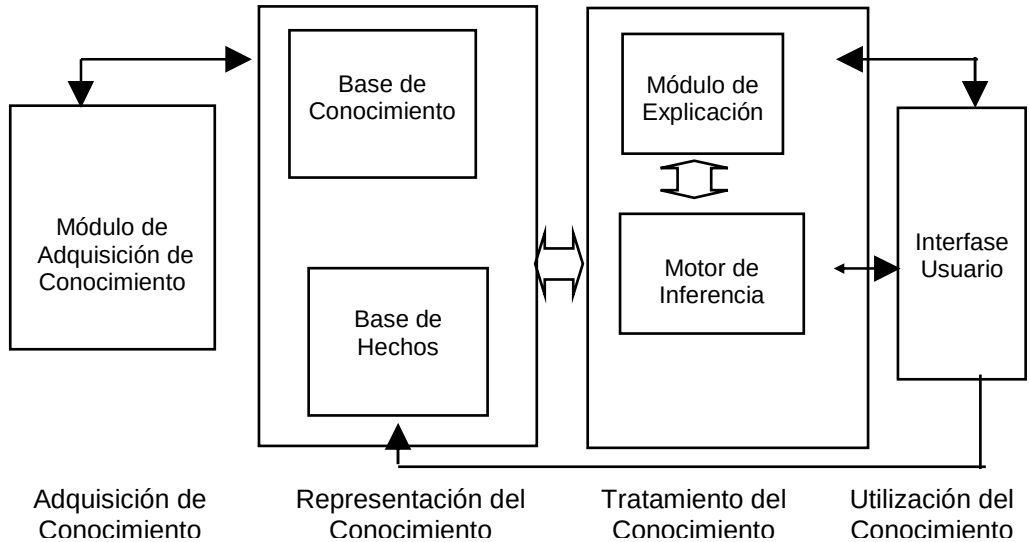


Figura 1: Arquitectura de un SE (Sistema Experto) [10], [18].

3. EL CASO DE ESTUDIO.

La FCACE presenta problemas en los procesos de ingreso de notas a su sistema académico, parte clave para los procesos de matricula, generación de cargas académicas, horarios, etc. Esta realidad es consecuencia directa de la falta de uso, desconocimiento de las herramientas informáticas, con lo cual se generan retrasos y deficiencias en la cadena de procesos principales para la institución [15].

Se propone desarrollar un sistema experto basado en el método de encadenamiento progresivo [21], como sistema de ayuda a los usuarios al utilizar las aplicaciones informáticas de la institución, este sistema utilizará premisas para reglas definidas previamente las cuales representaran los errores y las acciones a tomar [14], para ello definimos el tipo de usuario de la aplicación en base a los criterios de la tabla No. 2 , luego construiremos una base de conocimiento (serie de Reglas " Antecedente 1, Antecedente 2 → Consecuentes ") [19], una base de hechos (proporcionado de acuerdo al caso) y definiremos el algoritmo basado en encadenamiento progresivo.

Para el desarrollo del sistema experto se siguió la metodología planteada por Weiss y Kulikowski [25], por ello se platea el problema en la primera etapa, seguida de la búsqueda de expertos en el tema con lo cual se definieron las características para el sistema, se diseñaron las reglas de decisión y el algoritmo para el motor de inferencia, se eligió como herramienta de desarrollo el Power builder 9 por homogeneidad con las aplicaciones existentes, luego se desarrollo el prototipo y se realizaron las corridas respectivas del algoritmo antes de implementarse el software.

Usuario	Descripción
Inexperto	Usuario completamente inexperto en informática
Inexperto en la aplicación	Usuario completamente nuevo en la aplicación a utilizar y maneja algunos contenidos informáticos
Nivel Intermedio en la aplicación	Usuario con conocimiento en la aplicación
Experto en la aplicación	Tiene una amplia experiencia

Tabla No. 2: Clasificación de los usuarios [16].

3.1 Definición de las características del sistema.

En la tabla Nº 3 se plasman algunas características relevantes para el uso del Sistema que más adelante serán utilizados para las reglas de la Base del Conocimiento.

VARIABLE	CARACTERÍSTICAS
C1	1er intento para ingresar al Sistema
C2	2do intento para ingresar al Sistema
C3	3er intento para ingresar al Sistema
X	Usuario y Contraseña CORRECTA

Y	Usuario ó Contraseña INCORRECTA
W	Clic en el Sistema – Free.
C4	Cuando no accede al **registro** en un tiempo mayor ó igual a 30 seg.
C5	Cuando no accede a **Curso+Curricula y T-S** en un tiempo mayor ó igual a 40 seg.
C6	Los casilleros de las notas NO están llenas
C7	Los casilleros de las notas están llenas
C8	Se ha terminado de ingresar la nota del Ex. Final.
C9	No se ha terminado de ingresar la nota del Ex. Final.
C10	Ingreso al Sistema.
C11	Salida del Sistema
A	Cuando no ha accedido al Reporte de los alumnos con notas en un tiempo mayor ó igual a 120 seg.; y la cantidad de alumnos es menor a 20.
B	Cuando no ha accedido al Reporte de los alumnos con notas en un tiempo mayor ó igual a 200 seg.; y la cantidad de alumnos es mayor a 19.
Z	Después de 10 minutos (Desde que ingresó al Sistema).

Tabla Nº 3: Características en el Uso del Módulo de notas. Elaboración Propia.

Las ayudas (soluciones para el sistema de ayuda) asociadas a cada acción del usuario en el uso del Sistema, se muestran en el cuadro siguiente:

SOLUCIÓN	**EXPLICACIÓN**
S1	En el 2do intento, el Sistema mostrará una advertencia **"Bloqueo del Sistema para un 3er intento fallido"**.
S2	Cuando el tiempo excede a 30 segundos, automáticamente, le salga un aviso **"Recuerde que tuvo que hacer clic en Registro. Esta vez yo lo hago"** y el Sistema inmediatamente hará clic en Registro.
S3	Cuando el tiempo excede a 30 segundos, le salga un aviso **"Se excedió el tiempo de costumbre, llamado: Remember!! Haga clic en el Curso + Curricula; luego en T – S"**; y le muestra el listado de los alumnos.
S4	Cuando el usuario este en el registro de Notas, y los casillero tengan notas, el sistema hará clic (automático) mostrando el mensaje para que el Usuario **GUARDE ó GRABE**, evitando pérdida de información.
S5	Cuando termina de ingresar la nota del Examen Final, le mostrará un mensaje" **"Cuando se termina de ingresar la nota del examen final, hay que Calcular Promedios?** Haga clic en Promedios.

S6	Cuando el tiempo se excede: Preguntar **"¿Desea Imprimir?"**, si la respuesta es OK; automáticamente clic en Preliminar, Clic en Imprimir.; de lo contrario clic en CANCEL.
S7	Tiempo – Desea salir? Cuando tiene más de 10 minutos conectado al Sistema, mensaje **"¿Desea salir del Sistema?"** Si la respuesta es SI. , le mostrará un mensaje "Ud. está abandonando el Sistema …"

Tabla Nº 4: Soluciones en el Uso del Módulo de notas. Elaboración Propia

Lista de Consecuentes terminales. Definido en [11], [24] y [23].

LCT (S1, S2, S3, S4, S5, S6, S7)

3.2 Método de encadenamiento progresivo.

3.2.1 Ingreso de requerimientos (BC)

R1:	Si	C1, X	→C10	R9:	Si	C10, C7	→ S4
R2:	Si	C2, X	→C10	R10:	Si	C10, C8	→ S4
R3:	Si	C2, Y	→ S1	R11:	Si	S4, C8	→ S5
R4:	Si	C3, X	→C10	R12:	Si	S4, C9	→ W
R5:	Si	C3, Y	→C11	R13:	Si	C10, C9	→ W
R6:	Si	C10, C4	→ S2	R14:	Si	C10, S4, A	→ S6
R7:	Si	C10, C5	→ S3	R15:	Si	C10, S4, B	→ S6
R8:	Si	C10, C6	→ W	R16:	Si	C10, Z	→ S7

Las prioridades en orden ascendente son:

R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14 R15 R16

3.2.2 Algoritmo – Todas las Soluciones – Con prioridad

1. Leer (BC); Leer (BH); Leer (Sol);
2. S_disp ← True; Sol ← { };
3. Ordenar reglas.
4. Mientras S_disp
   S_disp ← False
   Verificar c/regla no disparada de BC
   Si (se dispara una regla Rx de consecuente Cx)
   Entonces S_disp → True, BH → BH + Cx
   Si Cx Є LCT Entonces Sol ← Sol + Cx;
5. Si Sol = { }
   Entonces Escribir ("No hay solución"),
   Sino Escribir ("Soluciones=", Sol);

Se utiliza este algoritmo todas las soluciones, porque existe la probabilidad de que el usuario necesite más de una solución (ayuda) en una sesión.

3.2.3 Corrida del Algoritmo y soluciones.

Para la corrida se tomaron en cuenta acciones realizadas por usuarios aleatorios en el uso del sistema de notas, con la cual generaron una base de hechos, se tomaron en cuenta 35 casos diferentes de los cuales se detallan a continuación 4 casos.

Caso 1:

Ite	Reglas no disparadas	Rx	Cx	BH	Sal	S_disp
0	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16	R2	C10	X, C2, C4	F	T /
1	R1, / , R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16	R6	S2	X, C2, C4, C10	F	F / T
	R1, / , R3, R4, R5, / , R7, R8, R9, R10, R11, R12, R13, R14, R15, R16			X, C2, C4, C10, S2	T	T

Solución : S2 Si BH => LCT
(X, C2, C4) => S2

Cuando el tiempo excede a 30 segundos, automáticamente, saldrá un aviso “Recuerde que tuvo que hacer clic en Registro. Esta vez Yo lo hago” y el Sistema lo hará de manera automática.

Caso 2:

Ite	Reglas no disparadas	Rx	Cx	BH	Sal	S_disp
0	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16	R1	C10	X, C1, C5	F	T /
1	/ , R2 , R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16	R7	S3	X, C1, C5, C10	F	F / T
	/ , R2 , R3, R4, R5, R6, / , R8, R9, R10, R11, R12, R13, R14, R15, R16			X, C1, C5, S3	T	T

Solución : S3 Si BH => LCT
(X, C1, C5) => S3

Cuando el tiempo excede a 30 segundos, le salga un aviso “Se excedió el tiempo de costumbre, llamado: “Remember!! Haga clic en el Curso + Curricula; luego en T – S”; y le muestra el listado de los alumnos.

Caso 3:

Ite	Reglas no disparadas	Rx	Cx	BH	Sal	S_disp
0	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16	R10	S4	C10, C8, B, Z	F	T /
1	R1, R2, R3, R4, R5, R6, R7, R8, R9, / , R11, R12, R13, R14, R15, R16	R11	S5	C10, C8, B, Z, S4	F	F / T
2	R1, R2, R3, R4, R5, R6, R7, R8, R9, / , / , R12, R13, R14, R15, R16	R15	S6	C10, C8, B, Z, S4, S5	F	F / T
3	R1, R2, R3, R4, R5, R6, R7, R8, R9, / , / , R12, R13, R14, / , R16	R16	S7	C10, C8, B, Z, S4, S5, S6	F	F / T
	R1, R2, R3, R4, R5, R6, R7, R8, R9, / , / , R12, R13, R14, / , /			C10, C8, B, Z, S4, S5, S7	T	T

Soluciones : S4, S5, S6, S7 Si BH => LCT

(C10, C8, B, Z) => S4, S5, S6, S7

- Cuando el usuario este en el registro de Notas, y los casillero tengan notas, el sistema hará clic (automático) mostrando el mensaje para que el Usuario GUARDE ó GRABE, evitando pérdida de información.

- Cuando termina de ingresar la nota del Examen Final, le mostrará un mensaje" "Cuando se termina de ingresar la nota del examen final, hay que Calcular Promedios? Haga clic en Promedios.

- Cuando el tiempo se excede: Preguntar "¿Desea Imprimir?", si la respuesta es OK; automáticamente clic en Preliminar, Clic en Imprimir.; de lo contrario clic en CANCEL.

- Time – Desea salir? Cuando tiene más de 10 minutos conectado al Sistema, mensaje "¿Desea salir del Sistema?" Si la respuesta es SI. , le mostrará un mensaje "Ud. está abandonando el Sistema…"

Caso 4:

Ite	Reglas no disparadas	Rx	Cx	BH	Sal	S_disp
0	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16	R8	W	C10, C6, A	F	T /
	R1, R2, R3, R4, R5, R6, R7, / , R9, R10, R11, R12, R13, R14, R15, R16			C10, C6, A, W	F	F / T

Solución : { } Si BH => LCT

(C10, C6, A) => { }

- La solución es el conjunto vacío, es decir, no existe solución.

4. SOFTWARE.

El software esta desarrollado en Power Builder 9 y tiene como característica principal que es orientado a objetos del mismo modo que la aplicación para el sistema de notas, se ha realizado el análisis y diseño correspondiente según las fases del desarrollo de un Sistema Experto [25] aunque se ha respetado también el método propuesto por Craig Larman [13] por la notación UML. El Sistema Experto presenta una interfaz de usuario bastante amigable, dependiendo de cada regla disparada por el usuario se desprende una ventana de ayuda de acuerdo al caso, la figura 2 muestra algunas pantallas captadas al utilizar el sistema, nótese que este es capaz de activar algunos controles si fuese el caso.

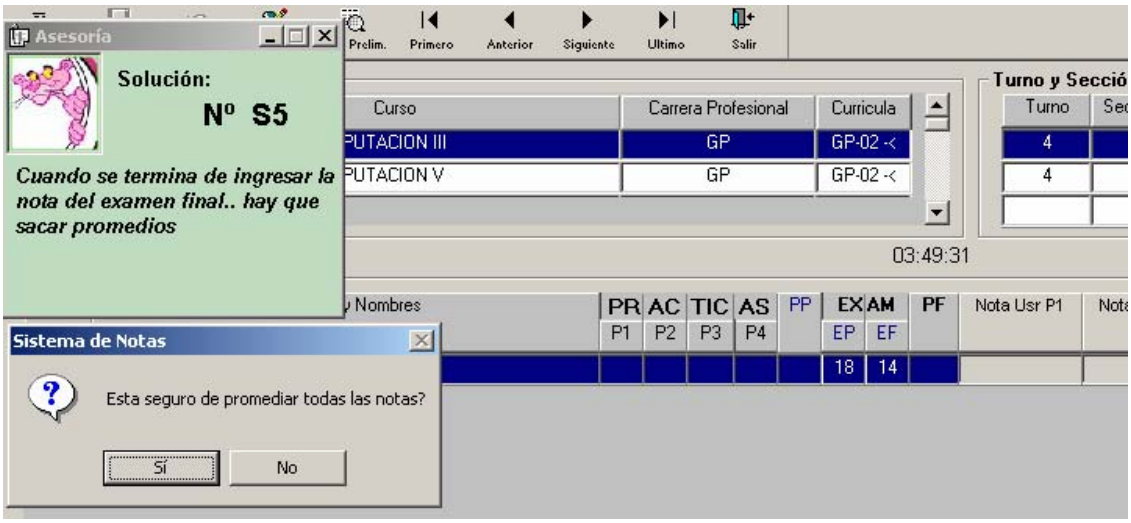


Figura 2. Pantalla mostrada por el sistema al dispararse la regla R11 y la acción correspondiente

5. CONCLUSIONES

Un sistema experto de ayuda proporciona apoyo o soporte en línea a los usuarios en el uso de aplicaciones informáticas con lo cual brindará un apoyo valioso a los especialistas del área.

El conocimiento del Sistema Experto se ha obtenido por experiencia y consulta a los especialistas, de una realidad vivida en los dos últimos semestres, donde la mayoría de usuarios presentaban inconvenientes en el uso y manejo del módulo de notas provocando sobrecarga en las labores de los encargados.

El sistema puede interpretar las acciones de los usuarios y determina la clase de ayuda que necesita incluyendo casos en los cuales activa controles que el usuario no realizó (vea solución S2), además con su incorporación se proporciona asistencia y enseñanza bajo demanda de los usuarios, con ello se espera mejorar su rendimiento en el uso de dichas aplicaciones, asimismo se demuestra que el algoritmo todas las soluciones con prioridad es el más adecuado para sistemas de ayuda de este tipo.

Referencias

[1]. B Wasson and Akselsen S, 1992. An Overview of On-line Assistance: from On-line Documentation to Intelligent Help and Training The Knowledge Engineering Review 7 (4) 289-322.

[2]. Baltasar Fernández Manjón. Sistemas de ayuda inteligente para entornos informáticos complejos. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.12 (2001), pp. 59-67. ISSN: 1137-3601. © AEPIA

[3]. Benchimol, G., et al. (1988) "Los Sistemas Expertos en la Empresa". Ra-Ma.

[4]. E. Castillo,J.M. Gutiérrez, and A.S. Hadi. Expert Systems and Probabilistic Network Models. Springer-Verlag, New York (1997).

[5]. Castillo Enrique, José Manuel Gutiérrez y Ali S. Hadi. Sistemas Expertos y Modelos de Redes Probabilísticas. Editorial Academia de Ingeniería (España). Madrid, España. 1998. ISBN 84-600-9395-6.

[6]. Durkin, J. (1994), Expert Systems: Design and Development. Maxwell Macmillan, New York.

[7]. Fernandez-Manjon, B., 1996. Desarrollo de sistemas de ayuda inteligente mediante integ[7]ración de tecnologías y reutilización de información. Tesis Doctoral, Universidad Complutense de Madrid.

[8]. Frye M J, Olynick D M, Pinkney R B. Development of an expert system for the fire protection requirements of the national building code of Canada. Construction Informatics Digital Library. Paper w78-1992-215.

[9]. Gervás, P., San Miguel, B. Un sistema experto basado en reglas para la automatización de la elaboración de horarios para un conjunto de restricciones particulares. III Jornadas de Transferencia Tecnológica de Inteligencia Artificial, Murcia, 16-19 noviembre 1999.

[10]. Giarratano Joseph, Gary Riley. Sistemas Expertos. Principios y Programación.. International Thomson. Editores (1998).

[11]. M. Henao Calad. CommonKADS RT: Una Metodología para el Desarrollo de Sistemas Basados en el Conocimiento de Tiempo Real. Tesis Doctoral. Universidad Polit Universidad Politécnica de Valencia 2002.

[12]. Kearsley, G., 1988. Online Help Systems: Design and Implementation. Ablex Publishing Corporation, Norwood, New Jersey, USA.

[13]. Larman. UML y Patrones. C. Prentice Hall, 1999.

[14]. Nevado A., C. de Mora, H. Pastor. Control Adaptativo Predictivo Experto: Metodología y Aplicación Industrial. I Seminario de Aplicaciones Industriales de Control Avanzado – SAICA 2005.

[15]. Paixão Jefferson, Cardoso; Claudia Ribeiro, Santos Lopes; Rodrigo Santos de Queiroz; Valéria Argolo Rosa; Alba Benemérita Alves Vilela. THE USE OF SPECIALIST SYSTEMS FOR SUPPORT TO ORGANIZATION IN ORTHOPEDICAL EXAMS OF THE HIP, KNEE AND ANKLE. Rev.Saúde.Com 2005; 1: 24-34

[16]. PALMER J. , DUFFY T. y MEHLENBACHER B. Online help systems: theory and practice. Ablex, Norwood, NJ, 1993.

[17]. Rios Ruiz Wilson Rafael. Propiedad Intelectual de las Obras creadas, producidas o

generadas mediante Ordenador (Los Sistemas de Inteligencia Artificial). Revista de Derecho Informático ISSN 1681-5726. Edita: Alfa-Redi-No. 052 - Noviembre del 2002

[18]. Salvador Ierache Jorge. Sistema Experto Aplicado al Control del Espacio Aéreo. Reportes Técnicos en Ingeniería del Software. Vol. 5(2). Pág. 41-91. 2003 ISSN: 1667-5002. © CAPIS-EPG-ITBA

[19]. Sierra, E., Hossian, A. y García-Martínez, R. Sistemas Expertos que Recomiendan Estrategias de Instrucción. Un Modelo para su Desarrollo. Revista Latinoamericana de Tecnología Educativa. Volumen 1 N° 1. Páginas 19-30. Facultad de Educación. Universidad de Extremadura. ISSN: 1695-288X. 2003

[20]. Soriano Zárate Omar, Antonieta Abud Figueroa. Gestión del Conocimiento utilización de XML- para bases de Conocimiento XML y derivados: Estándares para un paradigma de bases de Conocimientos -IST de Orizaba Veracruz – México. Energía y Computación-Edición 22

[21]. Sowa John F., Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, ©2000.

[22]. Sowa John F.. Architectures for intelligent systems," IBM Systems Journal 41:3, 2002, pp. 331-349.

[23]. C Tasso, G. Guida . Design and Development of Knowledge Based Systems. England: John Wiley & Sons. 1994. 476 p.

[24]. Waterman D.. A Guide to Expert. United States of America. 1986. 419p.

[25]. Weiss y Kulikowski, "SISTEMAS EXPERTOS", Prentice–Hall, 1984.

USO DE GRAFOS PARA EL MODELADO DE EXPERIENCIAS EDUCATIVAS COLABORATIVAS BASADAS EN LA WEB

Norma MORONI – Perla SEÑAS

[psenas/nem]@cs.uns.edu.ar
Laboratorio de Investigación y Desarrollo en Informática y Educación (LIDInE)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur - Bahía Blanca

Resumen

La educación basada en la Web introdujo nuevas variaciones en los modelos y supuestos de la educación formal. El aprender a aprender, las comunidades de aprendizaje, la formación continua, el aprendizaje autónomo, el colaborativo y la promoción de un auténtico interés en el alumno han adquirido una renovada relevancia. A ello se agrega la cognición e información situada, así como la inteligencia distribuida, procesos que permiten que solidariamente se aborde la identificación de problemas y la planeación y ejecución colectiva de las opciones más productivas de solución a los mismos. Todo ello presiona para la definición de un nuevo paradigma educativo en el que las Ciencias de la Computación y las Ciencias de la Educación tienen aún mucho que aportar. Permanentemente se buscan superaciones tecnológicas para poder diseñar y fundamentalmente mantener ambientes de aprendizaje personalizados, con contenidos y materiales apropiados para las aspiraciones y necesidades de cada uno. En tal sentido se presenta en este trabajo una forma de modelar, a través del uso de grafos, la organización de una comunidad educativa colaborativa mediada por la Web. Ello permite aprovechar conceptos y algoritmos propios de estas estructuras para la reorganización automática de la comunidad educativa que participa de la experiencia.

Palabras Clave: aprendizaje basado en la Web – aprendizaje colaborativo - modelado - grafos

1. Introducción

La educación formal, interpretada como acto de aprender, como acción y efecto de enseñanza y aprendizaje ha ido cambiando a través del tiempo. Se ha modificado la metodología de enseñanza basada en teorías psicopedagógicas que fueron progresando y evolucionando a través del tiempo. Esa evolución se produjo en forma continua. No obstante, se ha percibido desde hace unos años una decadencia progresiva en el resultado obtenido. Esta decadencia se ha adjudicado a distintos factores, no sólo educativos sino también sociales y económicos. La brecha entre la escuela y la sociedad se hizo cada vez mayor; mientras la escuela se queda en el tiempo la sociedad corre tras la vorágine de las tecnologías. Existe una nueva generación de personas que nace junto y por lo tanto familiarizada con el medio tecnológico. No sólo se piensa en los cambios a nivel escolar elemental, sino que el movimiento social hace que el mundo con posibilidades de aprender lo haga en forma permanente. Los avances en las ciencias y en las tecnologías de todo tipo hacen que los profesionales necesiten actualizarse, alcanzar nuevas competencias. Las tecnologías informáticas y de las comunicaciones (TICs) ofrecen posibilidades para facilitar el acceso al aprendizaje. Se pueden encontrar en la Web, muchísimos cursos que ofrecen formación para cada especialidad y portales que acercan las facilidades de la tecnología para lograr que el usuario acceda a la información, interactúe con otros usuarios y con docentes. No se presenta ya el e-learning como una mejora de la educación formal presencial, sino como una posibilidad de enseñar y de aprender distinta. Se constituye a veces como un complemento de la modalidad tradicional, en otras como una alternativa y hasta en ocasiones como la única opción posible, [1].

2. Experiencias educativas basadas en la Web

Cuando se plantea una experiencia educativa es importante comenzar identificando las entidades relevantes, tales como actores, objetos y relaciones. En la mayoría de los cursos on line, sobre todo cuando se busca conformar comunidades de aprendizaje, existe la necesidad de relacionar a cada alumno con:

- el material de estudio.
- el docente on line (DOL)
- el o los moderadores on line (MOL)
- uno o más alumnos on line (AOL) que realizan el curso

Cada AOL debe tener posibilidad de:

- Acceder al material de estudio en cualquier momento.
- Obtener evaluaciones continuas de su progreso
- Ampliar el tema desarrollado
- Elegir un grupo de trabajo
- Elegir a otro AOL para una interacción más intensiva

El DOL debe tener la posibilidad de:

- Determinar el material de curso
- Evaluar la interacción de los moderadores con los AOLs y de los moderadores entre sí.
- Monitorear y evaluar el curso estableciendo parámetros de éxito y fracaso.

El MOL debe tener la posibilidad de:

- Interactuar con los AOLs.
- Interactuar con los otros moderadores.
- Interactuar con el docente.
- Detectar posibles deserciones y aplicar estrategias para evitarlas.
- Evaluar continuamente la actividad, los momentos de comunicación, la calidad de intervención, los resultados obtenidos y el aprendizaje de cada alumno.

La educación formal a distancia es factible, pero tiene implícita una nueva forma de comunicación educativa que requiere de un análisis, revisión, evolución y generación de nuevas teorías psicopedagógicas que posibiliten un mejor aprovechamiento de los medios para lograr un beneficio significativo en los actos de enseñar y de aprender. La diagramación del curso es fundamental. Se debe contar con un entorno de software que permita al AOL conformar comunidades de aprendizaje, someterse a evaluaciones continuas que lo ayuden a percibir la construcción de su propio conocimiento. El avance del curso se deberá registrar para que los moderadores puedan realizar el seguimiento de cada AOL en particular y de la comunidad en general.

Todo se produce en línea, aún las comunicaciones asincrónicas. Éstas son tan importantes como las sincrónicas. Las comunicaciones asincrónicas no tienen una dependencia temporal, se pueden realizar en cualquier momento, pero son unilaterales. El e-interlocutor no siempre está presente en el instante en que se le es enviado un mensaje, como para responder inmediatamente. Las comunicaciones sincrónicas facilitan la comprensión y el entendimiento entre los pares, son bilaterales o multilaterales, pero deben realizarse en un intervalo de tiempo determinado. Resulta en muchos casos difícil compatibilizar horarios para todos los integrantes del grupo AOL. Actualmente, en un curso on-line, es habitual que se divida el grupo de manera tal que el conjunto de personas que intervienen sea lo suficientemente estable e interactivo para realizar el aprendizaje en forma cooperativa. Existen dos factores que deben tenerse en cuenta ya que favorecen el aprendizaje colaborativo, activo y participativo: la cardinalidad del grupo y la elección de los AOL

que lo conforman. Por lo general la selección de los grupos la proponen los moderadores, pero en algunas oportunidades, debieran poder realizarla los propios alumnos.
Existe otro problema no menor: el desgranamiento del curso. Un alto grado de deserción ocurre en los cursos a distancia basados en la Web. Es importante poder capturar la atención y mantener el compromiso entre todos. No obstante, se deben tener en cuenta los distintos aspectos causantes del abandono y luego aplicar las estrategias necesarias para evitarlo, [3]. Las causas de abandono del curso pueden tener su origen en el alumno, en el curso en sí y/o en los recursos tecnológicos a disposición:
Con respecto al alumno, se distinguen aspectos tales como:

- Falta de interés en el curso. Difícil de revertir.
- Falta de familiarización con la tecnología. Solucionable con un entrenamiento pre-curso.
- Sentimiento de abandono por parte de sus pares y/o de sus maestros.
- Falta de estímulo permanente en el progreso de su aprendizaje. Evaluaciones o auto evaluaciones que le confieran seguridad en el avance del estudio.
- Demora involuntaria en el seguimiento del curso. Evitable en muchos casos con recursos informáticos de mejor calidad.

En cuanto al curso en sí, se pueden considerar como las más destacables:

- Volumen excesivo de información. Evitable con evaluaciones previas del material del curso.
- Carencia de interacción con el material propuesto. Evitable con la incorporación de acciones al respecto.
- Escasa información al alumno de su progreso.

Los recursos tecnológicos tanto de hardware como de software pueden ser de:

- Pobre posibilidad de comunicación con sus maestros.
- Pobre posibilidad de comunicación con sus pares.
- Sin posibilidad de reinserción en el curso ante un atraso en el seguimiento del mismo. Se debe permitir la reformulación de grupos con los alumnos que se encuentren más atrasados en el seguimiento del curso.

3. Propuesta

Para realizar lo planteado es necesario en principio modelar un sistema que permita estructurar la información y la comunicación. A continuación se presenta el esquema de la comunicación que se establece entre las distintas entidades que intervienen en el acto educativo mediado por la Web. Por simplicidad, el modelo contempla sólo las relaciones que corresponden a la propuesta didáctica basada en el aprendizaje colaborativo.

3.1. Modelado

En este escenario educativo se presentan las siguientes categorías: *Docente, Moderador, Alumno, Ponencia* (expresión pública de un actor) *y Tema* (conjunto de ponencias con cohesión semántica). Estas categorías responden a una estructura jerárquica de comunicación y cada una de ellas está formada por una cantidad variable de actores u objetos que se relacionan entre sí. El esquema está basado en una estructura de digrafo jerárquico. Éste posee un nodo distinguido como raíz que representa al docente responsable del curso.
En el grafo se establecen cuatro tipos de relaciones que rotulan las adyacencias:

- *Comunicación* entre las personas que intervienen. Está indicada por un arco continuo bidireccional etiquetado con “se comunica con” (representado sin flechas). Esta relación se establece entre las tres primeras categorías mencionadas e interiormente en cada una de dichas categorías.

- *Producción* que corresponde a los mensajes que los AOL, los MOL y DOL publican. Está representado en la figura por un arco unidireccional etiquetado como “produce”.
- *Referencia* que corresponde a los mensajes que hacen referencia a otros propios o de otras personas. Estas relaciones están representadas por un arco unidireccional etiquetado como “referido a”.
- *Cronológica* que establece la secuencia de recepción de los mensajes. Es este caso se ha rotulado a los arcos con “precede a”.

Interactuando con el docente y en forma inmediata se encuentran los MOL, cuya cantidad dependerá de la cantidad de alumnos del curso (figura 1).

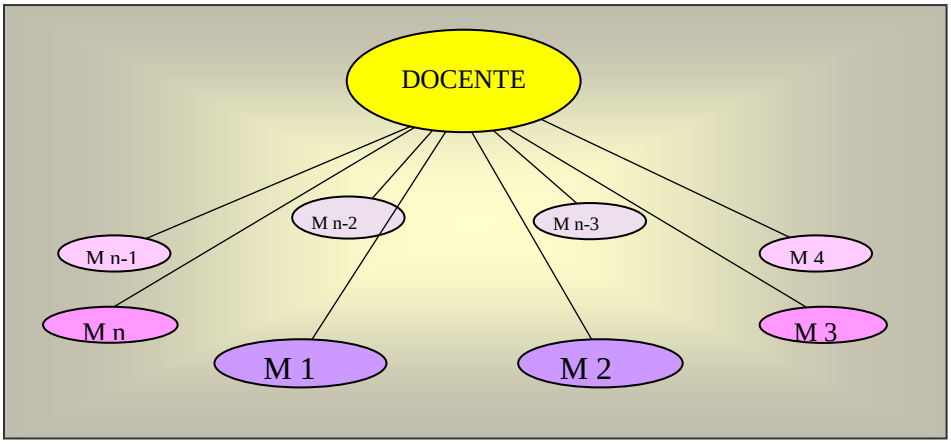


Figura 1

La comunicación del docente con los moderadores debe ser fluida, no necesariamente permanente sino diariamente posible. Pero también debe existir comunicación diaria entre los MOL participantes. La figura 2 sólo muestra la comunicación del moderador Mn con sus pares. El grafo debe representar la comunicación de cada moderador respecto de los restantes.

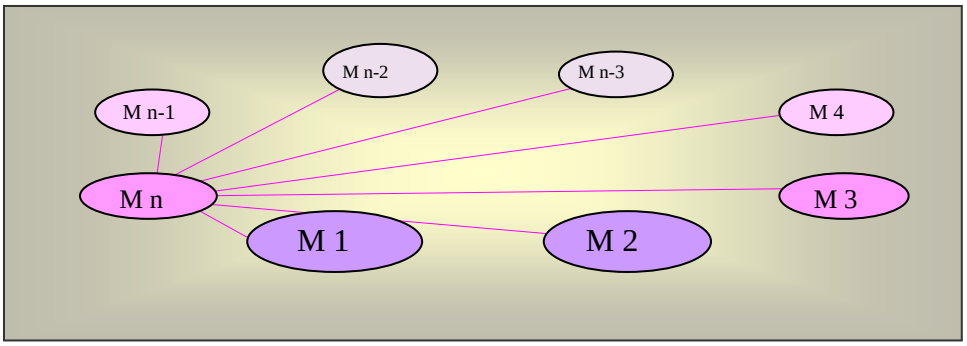


Figura 2

Cabe destacar que la comunicación es de todos a todos. Sólo por razones de claridad no se han representado en el gráfico. La figura anterior constituye la representación del grafo de la categoría de los MOL. Cada moderador tiene a su cargo un grupo de alumnos que interactúan con él y con sus pares, por lo tanto el grafo que representa las comunicaciones es similar al establecido entre el docente y los moderadores y entre los docentes entre sí. En forma similar se pueden representar las relaciones entre cada alumno con sus propias ponencias y las relaciones con las de sus pares, sólo que en estos casos los arcos son unidireccionales. En esta categoría se establecen los tres últimos tipos de relaciones que se han mencionado anteriormente.

Se puede observar en la figura 3 un sector del digrafo jerárquico cuyo nodo raíz es el Docente y a partir del cual se puede recorrer pasando por las distintas categorías hasta una ponencia de un alumno. De un nivel al inmediato siguiente se cambia de categoría (flechas negras). Dentro de cada categoría se establecen las comunicaciones entre los pares (flechas de color). Con el propósito de lograr una mejor visualización no se han incorporado en el grafo los descendientes de los M_i, ni las ponencias correspondientes a los alumnos A_j, como así tampoco las relaciones entre los MOLs M_k, con i,j,k>1. Para cada par de categorías inmediatas el grafo es similar, por lo que se puede pensar en un desarrollo recursivo del mismo. Las flechas negras parten de un nodo hacia el nodo descendiente y recíprocamente. Dentro de una misma categoría, existe otra relación entre los nodos. Se podría considerar como un grafo separado donde las relaciones se establecen entre todos los nodos del mismo. Dichas relaciones también permiten un desarrollo recursivo. Cada alumno que escribe un mensaje se encuentra relacionado con éste. Y dicha relación se representa en el grafo por un arco dirigido "produce". *Produce* es una relación de alumno a mensaje. Cada alumno puede escribir más de un mensaje.

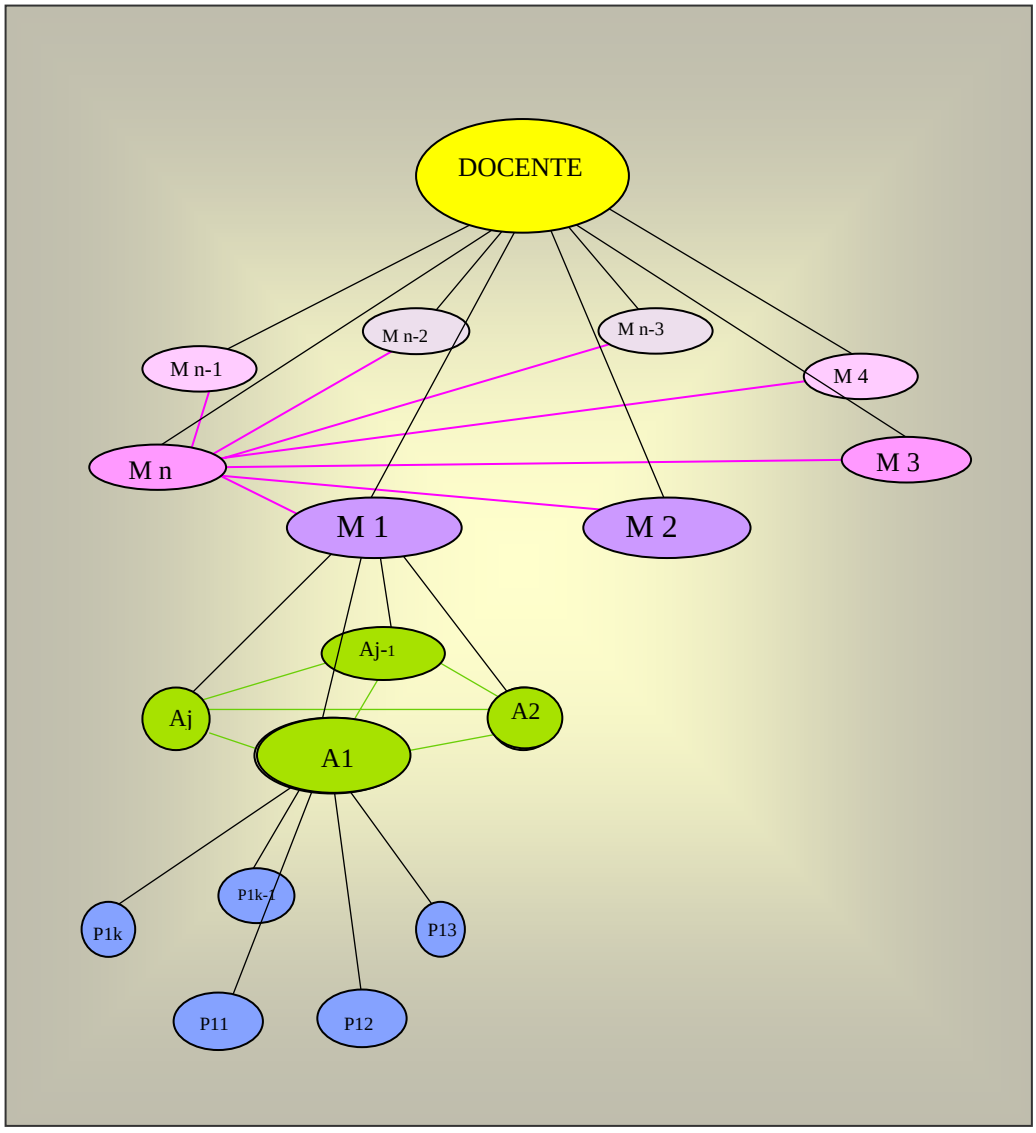


Figura 3

3.2. Categoría de los mensajes

En la clase de los mensajes o ponencias se produce un fenómeno distinto al que se produce en las otras categorías. Aquí se deben establecer relaciones que no existen en las anteriores donde los descendientes de un nodo se relacionan entre sí, pero son independientes de los otros descendientes de otro nodo. Por ejemplo, el conjunto de los alumnos de un moderador se relaciona entre sí, pero no necesariamente con el conjunto de alumnos de otro moderador. Esto no significa que no se permita la comunicación intergrupos, sino que dicha comunicación se realiza por otro carril que no está considerado aquí por ser irrelevante para el modelado que se presenta. Por ejemplo si se tratara de un foro, en principio, cada producción está relacionada con la siguiente respetando el momento en que son realizadas. Es decir, entre ellas hay un orden cronológico. Es más, existen dos órdenes cronológicos a tener en cuenta:

- El individual, que interesa fundamentalmente al tutor para llevar el registro del alumno, realizar estadísticas de la frecuencia de intervenciones, evaluar la calidad de cada intervención, etc. Se puede formar una cola de ponencias independientes de cada alumno en particular.
- El compartido en el que se encuentran involucradas todas las ponencias de los alumnos de un mismo grupo.

En la figura 4, A_i, con i = 1,2 se refiere a un alumno y P_{ij} a la ponencia j-ésima del alumno A_i. La flecha inferior representa en forma general las relaciones que deben existir entre ambos alumnos. Los arcos entre ponencias representan el orden cronológico individual en que se realizan las intervenciones de cada alumno. Las flechas punteadas representa la existencia de posibles ponencias.

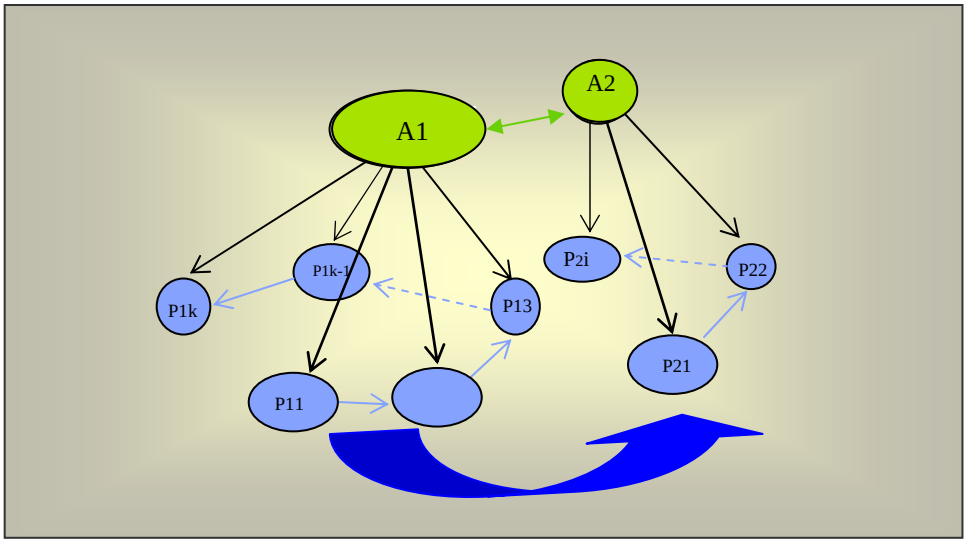


Figura 4

El orden cronológico de todas las ponencias del grupo de alumnos bajo un mismo moderador trasciende a cada alumno. En esta categoría, las ponencias están organizadas en una estructura de cola cronológica y son independientes del alumno que las formuló. Cada intervención se incorpora por un enlace cronológico a dicha cola, independientemente del autor. Esto obliga a realizar un acceso secuencial a las ponencias como lo ilustra la figura 5.

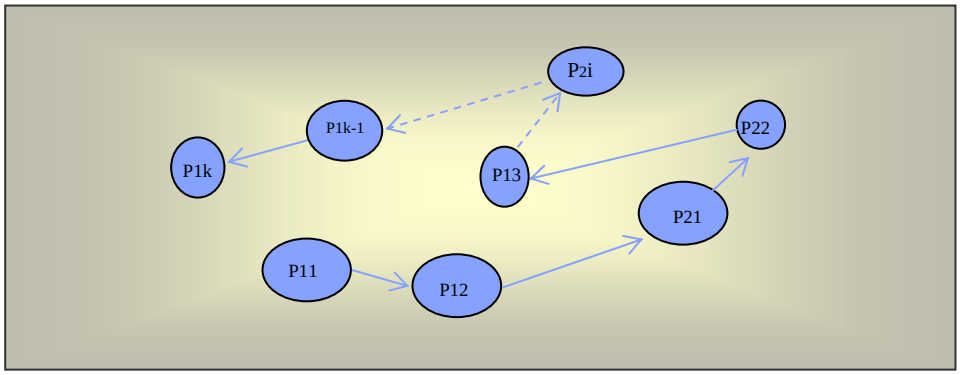


Figura 5

En la figura, el sentido de los arcos indica el orden cronológico en que se pueden producir las intervenciones. En este caso, las de los alumnos A_1 con P_{11}, P_{12}, P_{13}, P_{1K-1}, P_k y A_2 con P_{21}, P_{22}, P_{2i}, forman la cola cronológica P_{11}, P_{12}, P_{21}, P_{22}, P_{13}, …, P_{2i},…, P_{1k-1}, P_{1k}. Las flechas punteadas indican que puede haber otros nodos entre los nodos origen y destino que no están representados en la imagen.

Por otra parte, un alumno puede realizar comentarios esclarecedores, ampliatorios, con distintos puntos de vista referidos a un determinado tema expuesto en alguno de sus mensajes. Todas esas intervenciones sobre un mismo tema deben estar relacionadas entre sí para favorecer su comprensión también por parte de los otros alumnos y del moderador. En este punto, con la finalidad de producir un aprendizaje colaborativo, se observa que es necesario establecer relaciones no sólo entre las ponencias de un mismo alumno sino entre éstas y las ponencias de todos los alumnos que intervienen en el grupo. Es decir, que a nivel de la categoría de producciones en el grafo se deben establecer nuevos arcos entre los nodos que representan las referencias a otras ponencias. Estos arcos se rotulan como "*referencia a*". Es importante que además del enlace secuencial entre ponencias puedan establecerse enlaces referentes a determinadas ponencias. Cada alumno puede referirse a sus propias manifestaciones o a las manifestaciones realizadas por otro.

La ventaja fundamental en este tipo de estructuras es que se puede hacer referencia a lo manifestado por un alumno en un momento determinado sin necesidad de realizar el recorrido de toda la cola. Esto permite que un alumno pueda acceder y responder a lo manifestado por su par respecto de un tema estableciendo una relación directa a dicha manifestación. En consecuencia, la organización y disposición cronológica, y por lo tanto secuencialmente ordenada a lo largo del tiempo, es modificada por los vínculos de referencia que se establecen.

Dentro de la categoría de las ponencias, cada ponencia de cualquier alumno de un grupo debe poder referirse semánticamente a las ponencias de otro alumno del mismo grupo. En la figura 6 se indican en azul las relaciones de referencia entre la ponencia P_{21} del alumno A_2 y todas las ponencias del alumno A_1.

En forma análoga debiera permitirse la referenciación entre las ponencias de todos los alumnos y entre las del mismo alumno. Si bien se puede pensar como la relación entre dos autores, en realidad se está estableciendo una relación entre ponencias. De una ponencia de un alumno a otra ponencia de otro alumno o bien del mismo alumno.

No necesariamente en un acto de aprendizaje serán usados todos los canales de comunicación, sino que debe permitirse el acceso directo a los mismos frente a una necesidad. Si sólo se producen los enlaces cronológicos entre las ponencias, solamente existe la posibilidad de acceder a ellas en forma secuencial lo que significa que la respuesta a una de ellas figure alejada de la misma en la secuencia. Esto insume excesivo tiempo y esfuerzo para aunar los conceptos de interés común.

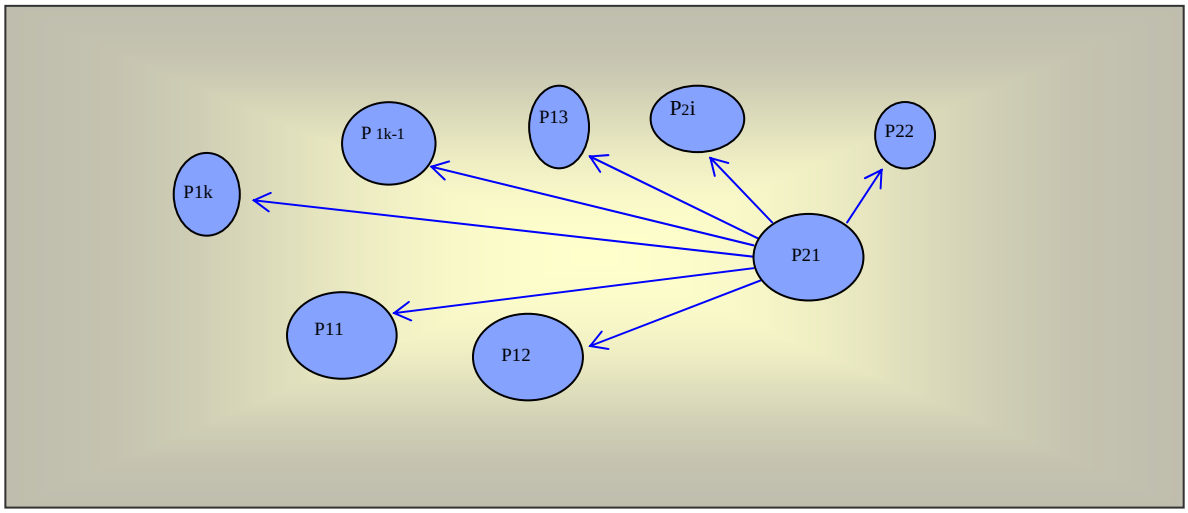


Figura 6

En principio para lograr la organización mencionada es interesante considerar la propuesta de Vitturini [22] para realizar la inclusión y la búsqueda de la información. De esta forma se puede referir a una ponencia que se encuentra suficientemente alejada en el tiempo y entre las que puedan existir intervenciones referidas a otras cuestiones. Por ejemplo, si P_{11} realiza una manifestación que es respondida por P_{22} y P_{13}, y además P_{1k-1} y P_{1k} se manifiestan sobre P_{2i}, se pueden establecer nuevos enlaces que reflejan más auténticamente el interés de los participantes del grupo, como lo muestra la figura 7.

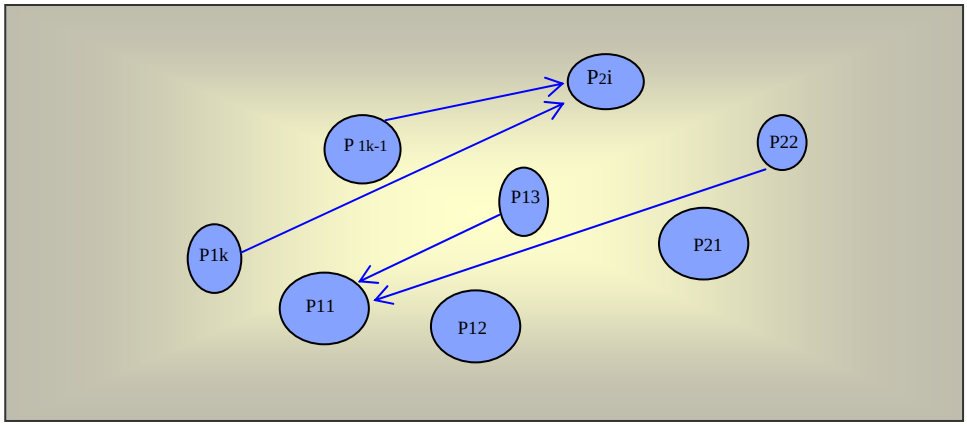


Figura 7

Como se puede observar en la figura, existen concentraciones de enlaces de referencia entre determinados nodos que representan los temas de discusión o de concordancia, y también, nodos aislados que reflejan el desinterés de los actores. Llamaremos foco de interés al subgrafo o subcola que alberga a las concentraciones de arcos y nodos. En el gráfico las ponencias P_{11}, P_{13} y P_{22} forman un foco de interés, como así también las ponencias P_{2i}, P_{1k-1} y P_k, mientras que las ponencias P_{12} y P_{21} constituyen focos aislados. La idea es que el sistema permita establecer referencias de todos a todos y que luego se especifiquen las que se necesitan. No se debe perder de vista que la cola cronológica se encuentra subyacente en esta categoría y mantiene ligado a todos los nodos

ponencias. La siguiente figura representa el ejemplo anterior indicando las relaciones de referencia en color azul y en forma solapada la cronológica.

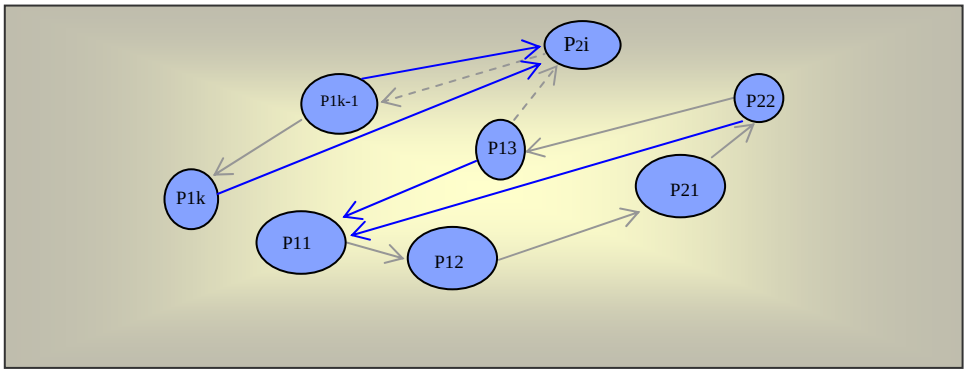


Figura 8

Focos temáticos y reorganización de los grupos de trabajo

A partir de la cola cronológica original, se pueden establecer subcolas, cada una de ellas formada por las ponencias que se refieren a un mismo tema. Cada subcola mantiene un enlace cronológico pero también establece el enlace de las respuestas, ampliaciones, aclaraciones, etc., con unidad semántica. Estas subcolas se forman estableciendo un nuevo enlace cronológico por transitividad lo que permite su acceso secuencial. En el gráfico (figura 9) se encuentran encerradas por elipses las dos subcolas que quedan formadas aplicando enlaces transitivos en la cadena cronológica y los enlaces de referencia. De esta manera se pueden detectar conceptos o temas subyacentes que resultan de interés a los alumnos y que generan una mayor actividad participativa y sobre los que se pueden centrar futuras discusiones, como así también aquellos conceptos que han quedado marginados.

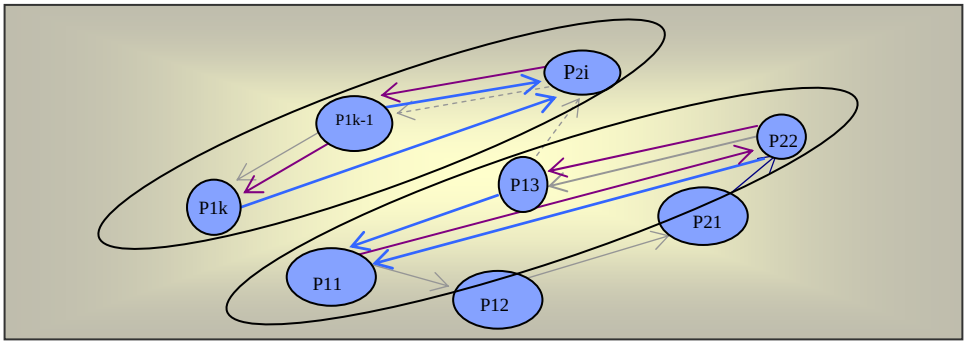


Figura 9

Frecuentemente uno de los problemas principales que se presenta en los cursos e-learning es su desgranamiento. Se produce una deserción por parte de los alumnos que lejos de ser natural parece inducida por la sensación de abandono y de distanciamiento con sus pares. Un alumno por determinadas circunstancias puede lentificar su ritmo colaborativo, lo que no indica que pretenda ausentarse definitivamente. Es importante que el sistema permita reinsertarlo en grupos en los que los alumnos se encuentren en la misma condición. Este modelado usando grafos tiene la ventaja de contar con algoritmos que pueden ser usados para la tarea de replanificación, [4] [7].

Analizando la cola de intervenciones de cada alumno el moderador puede establecer la frecuencia y la calidad de las mismas. Es decir, se necesita saber el número de intervenciones y cuando las ha realizado para tener un panorama claro del momento donde comienza a decaer su ritmo. Por otra

parte, es importante analizar la calidad de las ponencias. Con las subcolas se puede analizar si ha colaborado en temas de interés para el grupo, si ha intervenido en focos aislados que sólo han reflejado sus ideas pero que no conciernen al resto, si solamente estableció una subcola con sus propias ponencias o bien si sus intervenciones fueron escasas pero de mucho valor para el grupo.
La información mencionada anteriormente se puede obtener a partir del digrafo jerárquico cuya categoría de ponencias refleja no sólo la frecuencia de intervenciones de cada alumno por el número de arcos del alumno a sus ponencias, sino que con las relaciones cronológicas y de referencia se puede detectar en parte la calidad de las mismas [12].
Se aprecia que el recorrido en la dirección docente-moderador-alumno-ponencias en la estructura, permite analizar tanto lo propuesto por cada uno de los alumnos en un tema particular como la puesta en común del grupo en ese tema. La búsqueda longitudinal puede representarse en un grafo con una o más hojas. Esta búsqueda es muy útil cuando se está monitoreando la actividad de un moderador por parte del docente o la de un alumno por parte del moderador o del docente, incluso la de todo el grupo. Este monitoreo es adecuado para estudiar el avance en la lección, el análisis de la intervención de cada alumno, la detección de abandono, la interacción entre los integrantes del grupo, la orientación frente a desvío, etc. Además es importante poder hacerlo por subtemas, sin necesidad de atacar el problema, si es que lo hubiera, cuando ya se ha tratado en su totalidad.

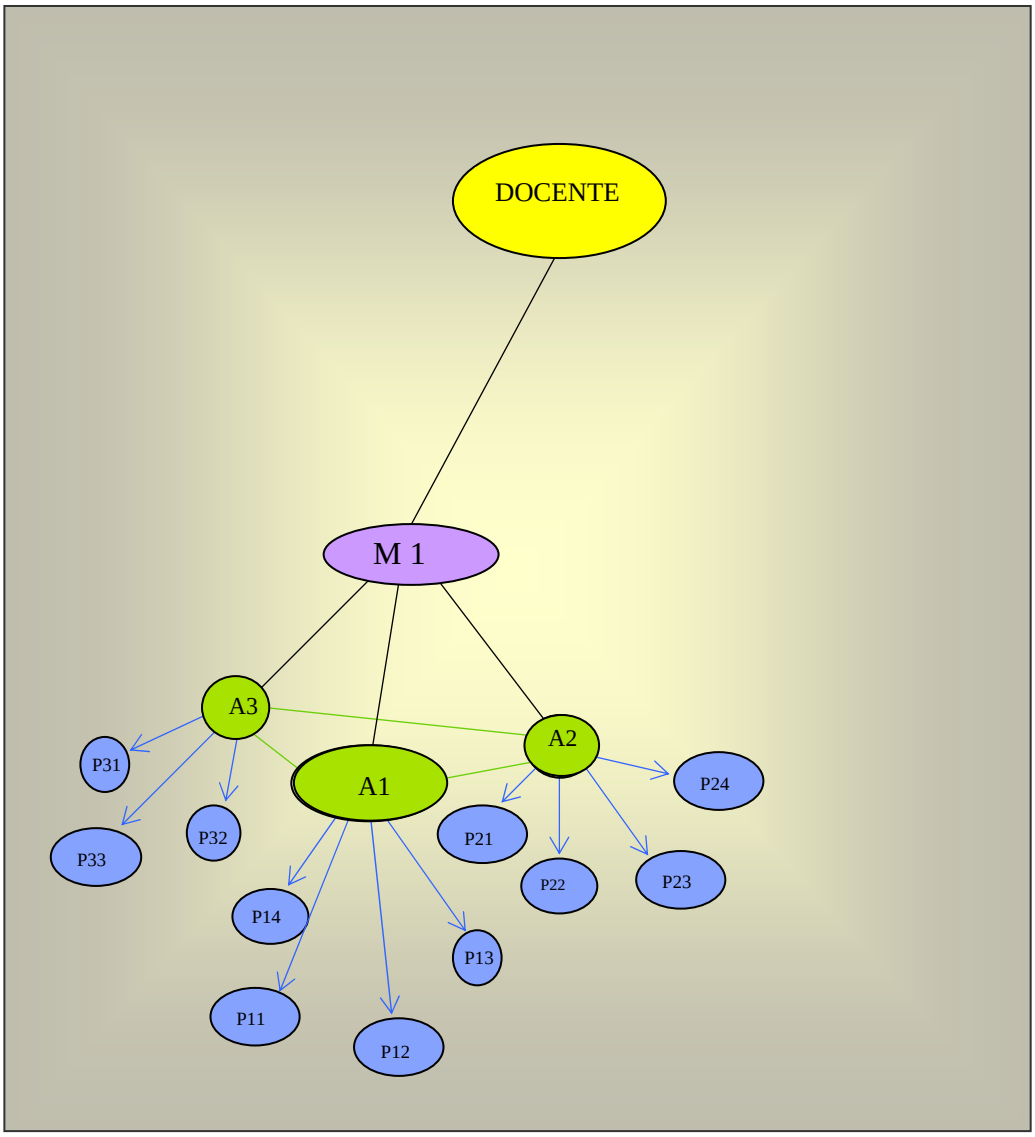


Figura 10

En la última figura se representa el plano de la categoría de las ponencias exclusivamente. Cada una de ellas identifica a su autor por el primer índice indicado en su rótulo, y la cronología individual por el segundo índice.

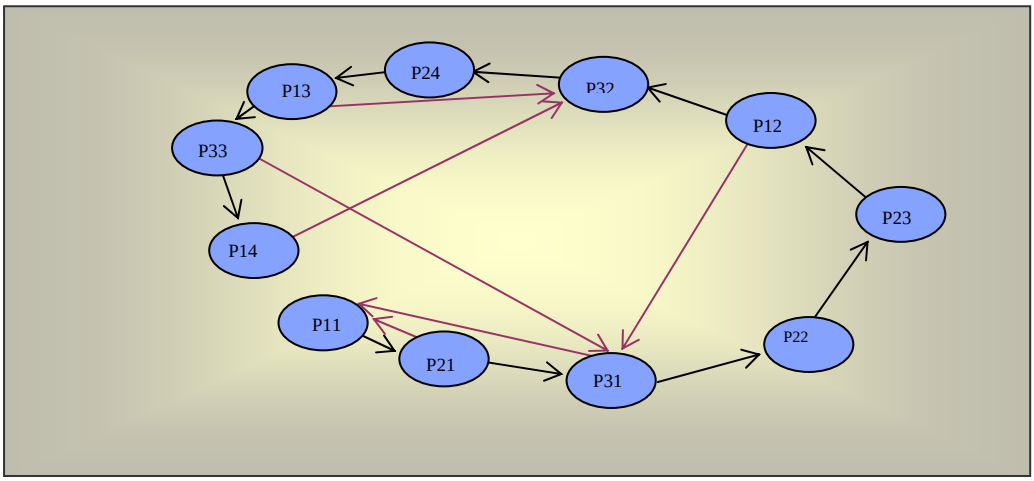


Figura 11

Con el fin de permitir una mejor visualización se reordenan los nodos y se establecen las relaciones de cronología general y referencia entre ellos.
La detección de los alumnos con problemas resulta muy sencilla teniendo en cuenta la representación propuesta.
Como se puede observar en la figura los alumnos A_1 y A_3 realizaron un trabajo colaborativo centrando sus discusiones en dos focos o temas principales mientras que el alumno A_2 si bien tuvo una fluida participación, sus intervenciones rara vez estuvieron referidas a alguno de los temas tocados por los otros alumnos. La generación de focos aislados indica que se está desviando del rumbo propuesto por el grupo y es necesario encausarlo.
Por otra parte, analizando las intervenciones del alumno A_1 se podría haber pensado en primera instancia que era un candidato a abandonar el curso, sin embargo sus posteriores intervenciones referidas a los temas de interés demostró lo contrario. De demorarse sus intervenciones en el tiempo A_1 y A_2 deberían ser reacomodados en grupos de demorados, que trabajarán a un ritmo distinto que el original.

4. Conclusiones

La educación basada en la Web introdujo variaciones en los modelos de la educación formal, se adaptaron algunos esquemas y se revalorizaron otros. Es así como muchas de las propuestas se inscriben en un paradigma colaborativo, activo y participativo. Diagramar una propuesta de este tipo no es una tarea trivial, además requiere de sucesivas replanificaciones, según vayan evolucionando los aprendices y de acuerdo a cómo se producen las comunicaciones entre los actores de la comunidad que se ha conformado. En este marco, contar con una forma de modelar el conjunto de comunicaciones posibles resulta alentador si se piensa que se hace en términos de una estructura conocida (el grafo), para cuyo manejo existen numerosos algoritmos que pueden permitir implementar la automatización de las mencionadas replanificaciones.

Bibliografía

[1] Andrej, K. and Bieliková, M. Improving adaptation in web-based educational hypermedia by means of knowledge discovery. Proceedings of the sixteenth ACM conference on Hypertext and hypermedia. 2005.
[2] ATutor. Información disponible en el sitio oficial (http://www.atutor.ca)
[3] Banzatto, M., Corcione D. "La tutoría en red-Funciones del Tutor on-line". Tutor on-line. Universidad de Salamanca. 2006.
[4] Brandenburg, F. J. Designining Graph Drawings by Layout Graph Grammars, Graph Drawing (Proc. GD ´94), vol. 894 of Lecture Notes Comput. Sci., Springer-Verlag, 1995.
[5] Brito, Vivina. "The electronic forum: a technological tool that fosters cooperative learning" Edutec. Revista Electrónica de Informática Educativa. Marzo 2004.
[6] Brusilovsky, P. Authoring Tools for Advanced Technology Learning Environments, chapter Developing Adaptive Educational Hypermedia Systems: from Design Models to Authoring Tools, pages 377-409. Kluwer Academic Publishers, 2003.
[7] Carpano, M. Automatic Display of Hierchized Graphs for Computer Aided Decision Analysis, IEEE Trans. Syst. Man Cybern., SMC-10, no. 11, 1980.
[8] Catarci, T. The Assigment Heuristic for Crossing Reduction in Bipartite Graphs, Proc. 26th. Allerton Conference Commun. Control Comp., 1988
[9] Engelhardt, M., A. Hildebrand, D. Lange, and T.C. Schmidt. Semantic Overlays in Educational Content Networks. in TERENA Networking Conference. Catania, Italy. 2006.
[10] Ka-Ping Yee, Danyel Fisher, Rachna Dhamija, Marti Hearst, Animated Exploration of Dynamic Graphs with Radial Layou.t. University of California, Berkeley.USA. In NFOVIS '01. Proceedings of the IEEE Symposium on Information Visualization 2001. IEEE Computer Society
[11] Landauer, T., Laham, D. and Derr, M.. From Paragraph to Graph: Latent Semantic Analysis for Information Visualization. Proceedings of the National Academy Sciences. 2004.
[12] Lin, T. y Eades, P. Integration of declarative and algorithmic approaches for layout creation. Graph Drawing (Proc. GD ´94), vol. 894, Lectures notes in Computer Science, Springer-Verlag, 1995.
[13] Millard, D., F. Tao, K. Doody, A. Woukeu, and H. Davis, The Knowledge Life Cycle for e-learning. International Journal of Continuing Engineering Education and Lifelong Learning: Special Issue on Application of Semantic Web Technologies in E-learning, 2006.
[14] Mochizuki, T., Fujitani, S., Isshiki, Y., Yamauchi, Y., Kato, H. (2003). Assessment of Collaborative Learning for Students: Making the State of Discussion Visible for their Reflection by Text Mining of Electronic Forums. Proceedings of E-Learn 2003 (Phoenix, AZ, November 2003), AACE, 285-288.
[15] Mutzel, P. An Alternative Method to Crossing Minimization on Hierarchical Graphs.
[16] Nakahara, J. and Nishimori, T. (Eds.) e-Learning: New Challenges of the Japanese universities (in Japanese). Ohmu Publishing Co., Tokyo. 2003.
[17] Palloff, R. and Pratt, K. (1999). Building Learning Communities in Cyberspace: Effective Strategies for the Online Classroom. Jossey-Bass, Inc. Pub.
[18] Sancho- Martínez- Fernández-Manjón. Semantic Web Technologies Applied to e-learning Personalization in e-aula. Journal of Universal Computer Science, 11(9): p. 1470-1481. 2005.
[19] Sheard, J. Electronic learning communities: strategies for establishment and management. Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education ITiCSE '04, Volume 36 Issue 3. 2006.
[20] Shirouzu, H., Miyake, N., and Masukawa, H. Cognitively Active Externalization for Situated Reflection, Cognitive Science, 26, 4, 469-501. 2002.
[21] Stratakis, M., V. Christophides, K. Keenoy, and A. Magkanaraki, E-Learning Standards. 2003.
[22] Vitturini-Señas. Semántica de los mensajes de un foro. CACIC- 2006.

Análisis de la Robustez del Método de Asignación MATEHa*

Laura De Giusti [1], Franco Chichizola [2], Marcelo Naiouf [3], Armando De Giusti [4]
{ldgiusti,francoch,mnaiouf,degiusti}@lidi.info.unlp.edu.ar

Instituto de Investigación en Informática (III-LIDI) – Facultad de Informática – UNLP

Abstract

The TTIGHa model has been developed to model and predict the performance of parallel applications run over heterogeneous architectures.

In addition, the task assignment algorithm was implemented to MATEHa processors based on the TTIGHa model.

This paper analyzes the assignment algorithm robustness before different variations which the model parameters may undergo (basically, communication and processing times).

Keywords: *Parallel Systems. Cluster and Multi-cluster Architectures. Performance prediction models. Tasks to processors mapping. Heterogeneous Processors Robustness.*

Resumen

Se desarrolló el modelo TTIGHa utilizado para modelizar y predecir performance de aplicaciones paralelas que se ejecutan sobre arquitecturas heterogéneas.

Además, se implementó el algoritmo de asignación de tareas a procesadores MATEHa basado en el modelo TTIGHa.

En este trabajo se analiza la robustez del algoritmo de asignación frente a diferentes variaciones que pueden sufrir los parámetros del modelo (básicamente tiempos de comunicación y tiempos de procesamiento).

Palabras Clave: *Sistemas Paralelos. Arquitecturas de Cluster y Multicluster. Modelos de predicción de performance. Mapeo de tareas a procesadores. Heterogeneidad. Robustez.*

WPDP – Workshop de Procesamiento Distribuido y Paralelo

---

[1] Becaria de Formación Superior UNLP. Profesor Adjunto Facultad de Informática UNLP.
[2] Becario de Doctorado del CONICET. Profesor Adjunto Facultad de Informática UNLP.
[3] Profesor Titular D.E. Facultad de Informática UNLP.
[4] Investigador Principal CONICET. Profesor Titular D.E. Facultad de Informática UNLP.

* *Esta investigación es financiada por la CIC , la Fundación YPF y el proyecto Grid CyTED.*

Introducción

En la Ciencia Informática, los modelos de computación son usados para describir entidades reales tales como arquitecturas de procesamiento y resultan una versión "abstracta" o simplificada de la máquina física, capturando características esenciales e ignorando detalles sin importancia de la implementación [1][2][3]. Un modelo no se relaciona necesariamente con ninguna computadora real, sino que su principal razón de ser es ayudar a comprender la computación. Provee un marco para estudiar problemas, obtener ideas sobre sus distintas estructuras, y desarrollar soluciones. Una vez que un algoritmo fue diseñado para resolver un problema con un cierto modelo, éste permite dar una descripción significativa del mismo, derivar un análisis detallado e incluso predecir la performance [4] [5].

En el caso de computadoras paralelas, los requerimientos mínimos que debe cumplir un modelo es ser conceptualmente simple de entender y usar, que la determinación de corrección de un algoritmo sobre el modelo sea válida independientemente de la arquitectura física, que *la performance real se corresponda con la predicha por el modelo,* y que se aproxime a las arquitecturas reales para minimizar la brecha conceptual entre modelo y arquitectura física.
En estos requerimientos queda claro que un objetivo central de los modelos de cómputo paralelo es la posibilidad de *predicción de performance* que brinden: el éxito o fracaso dependerán en gran parte de este punto [6].

Actualmente las arquitecturas más utilizadas por su relación costo/performance son los clusters y multiclusters de procesadores, razón por la cual resulta de importancia el desarrollo de un modelo que se adecue a las características de estas plataformas. Un elemento fundamental que aparece en estas arquitecturas es la potencial *heterogeneidad* de los procesadores y las comunicaciones entre los mismos, lo cual agrega un elemento a la complejidad intrínseca de la modelización. [7][8].

En la actualidad existen diferentes modelos basados en grafos para caracterizar el comportamiento de aplicaciones paralelas en arquitecturas distribuidas [9][10][11]. Entre los modelos se pueden mencionar el modelo TIG (Grafo de Interacción de Tareas), TPG (Grafo de Precedencia de Tareas) y TTIG (Grafo de Interacción Temporal de Tareas) [12].

El problema de los modelos mencionados anteriormente es que consideran que la arquitectura en donde se ejecuta la aplicación es homogénea, situación que en general no se da en clusters y multiclusters. Por ello se ha desarrollado el modelo TTIGHa el cual considera la heterogeneidad tanto en los procesadores como en la red de comunicación [13].

Una vez definido el grafo que modela la aplicación, el problema de "mapping" se resuelve mediante algún algoritmo que establece un mecanismo automático para realizar la asignación de tareas a procesadores, y así obtener mejores resultados en la ejecución de la aplicación [14] [15] [16] [17]. Este es un problema NP-completo, debido a la existencia de gran cantidad de factores a tener en cuenta que directa o indirectamente influyen en el tiempo de ejecución del programa. Los algoritmos de mapping estático pueden clasificarse en dos grandes grupos:

- *óptimo:* se evalúan todas las posibles formas de asignar las tareas a los diferentes procesadores. Este tipo de soluciones solo puede abordarse cuando el número de configuraciones posibles es lo suficientemente bajo. En caso contrario la solución óptima no puede llevarse a cabo debido a la explosión combinatoria en el número de soluciones posibles.

- *heurístico:* se basan en técnicas de aproximación que utilizan suposiciones "realistas" del algoritmo y sistema paralelo. Dichos algoritmos producen soluciones subóptimas en tiempos de ejecución más razonables comparado con las estrategias óptimas.

Para este trabajo se utiliza el modelo TTIGHa para representar las aplicaciones a ejecutar, y el algoritmo MATEHa para resolver el problema de la asignación de los procesos a los procesadores que forman la arquitectura antes mencionada.

Modelo TTIGHa

El modelo TTIGHa se basa en la construcción de un grafo G (V,E) para representar la aplicación que se quiere modelar. Para la construcción de dicho grafo se utilizan, además de información de la aplicación, parámetros que permiten caracterizar la arquitectura (Tp,Tc), donde Tp es el conjunto de procesadores y Tc representa el conjunto de tipos diferentes de comunicación.

Los elementos que componen el grafo son:

- V, es el conjunto de nodos donde cada uno representa una tarea T_i del programa paralelo.
- E, es el conjunto de aristas que representan la comunicación entre los nodos del grafo.

Detalle de los parámetros del modelo

Tp involucra al conjunto de procesadores. Como se mencionó en la definición de la arquitectura la misma puede ser heterogénea, por lo tanto se tiene un conjunto de tipos diferentes de procesadores, y cada elemento del conjunto Tp debe indicar a qué tipo de procesador pertenece.

Tc representa al conjunto de tipos diferentes de comunicaciones. Para cada comunicación del conjunto se indica su tiempo de startup y de transferencia de un byte; esto es necesario ya que el modelo permite que la red de interconexión sea heterogénea.

En el primer parámetro del grafo (V) cada nodo representa una tarea T_i del programa paralelo. En cada nodo se almacena el tiempo de ejecución correspondiente a la tarea que representa en cada tipo de procesador.

En el segundo parámetro del grafo (E), las aristas representan cada una de las comunicaciones que existen entre cada par de tareas. En este conjunto una arista A entre dos tareas T_i y T_j mantiene una matriz C de dimensión [**m x m**] (**m:** cantidad de procesadores de la arquitectura), donde $C_{ij}(s,d)$ es el tiempo de comunicación entre la tarea T_i ubicada en el procesador s y la tarea T_j alocada en el procesador d. Es importante notar que el costo de comunicación es dependiente de los procesadores que se comunican ya que la red de interconexión se considera heterogénea. Además mantiene el "grado de concurrencia" entre la tarea T_i y la tarea T_j .

El "grado de concurrencia" es una matriz h_{ij} de dimensión [**m x m**], donde $h_{ij}(s,d)$ representa el grado de concurrencia entre las tarea T_i en el procesador s y la tarea T_j en el procesador d. Este índice (grado de concurrencia) está normalizado entre 0 y 1. Para dos tareas T_i y T_j que se comunican de T_i a T_j, el grado de concurrencia se define como el máximo porcentaje del tiempo de cómputo de T_j que puede ser realizado en paralelo con T_i, teniendo en cuenta sus dependencias mutuas provocadas por las comunicaciones existentes entre ambas tareas, y sin contemplar el costo de comunicación asociado a las mismas (esto genera un valor independiente a los datos a transmitir).

Algoritmo MATEHa

Este algoritmo permite determinar la asignación de tareas a los procesadores de la arquitectura a utilizar buscando minimizar los tiempos de ejecución de la aplicación en dicha arquitectura. MATEHa considera una arquitectura con un número acotado de procesadores, que pueden ser heterogéneos en cuanto a su potencia de cálculo y a la red de interconexión.

La estrategia de MATEHa consiste en determinar, para cada una de las tareas del grafo G formado por el modelo TTIGHa, a qué procesador debe ser asignada para lograr el mayor rendimiento de la aplicación en la arquitectura utilizada. Dicha asignación usa los valores generados en la construcción del grafo: tiempo de cómputo de una tarea en cada procesador, tiempo de comunicación con sus adyacentes (el cual también depende de donde han sido asignadas las tareas) y por último el grado de paralelismo entre tareas. Este último valor es útil para tomar la decisión de asignar al mismo procesador aquellas tareas con menor grado de paralelismo, o asignar a procesadores diferentes aquellas tareas que pueden ejecutarse en forma concurrente (el valor de su grado de paralelismo es alto).

El algoritmo de mapping extrae los valores mencionados anteriormente del modelo TTIGHa, en los que se fundamenta la heurística de asignación del algoritmo. En primer lugar, para cada nodo del grafo del modelo TTIGHa se define el *nivel* que será el que se utilice para realizar la asignación de las tareas del grafo con cierta prioridad.

En segundo lugar, para cada nivel n del grafo (comenzando del nivel 0), se realiza la asignación de todas sus tareas a los procesado. Para esto, en cada paso se elige aquella tarea aún no asignada perteneciente al nivel n que genera la máxima ganancia al asignar dicha tarea a un procesador. La ganancia de una tarea Ti se obtiene como la diferencia entre el costo de ejecutar Ti en el “peor procesador” y la ejecución de Ti en el “mejor procesador” (esto no implica que el mejor / peor procesador sea el más rápido / lento respectivamente).

Para calcular el costo c de ejecutar una tarea Ti en un procesador p se realizan dos acciones. La primera suma al tiempo acumulado en p (este tiempo es la suma de los tiempos de ejecución de las tareas ya asignadas a él) el tiempo requerido para ejecutar Ti en p. En la segunda, por cada tarea Ta adyacente a Ti, que ya ha sido asignada a un procesador q (diferente a p) se acumula al costo c tiempo de comunicación entre Ti y Ta (en ambos sentidos) y el tiempo en que Ti y Ta no pueden ejecutarse de manera conjunta, es decir, el porcentaje en que no se ejecutan concurrentemente (1-grado de concurrencia entre Ti y Ta) multiplicado por el tiempo de ejecutar Ta en q.

Contribución de este trabajo

En el diseño de algoritmos automáticos de mapping, una característica importante es su robustez, ya que los algoritmos robustos permiten encontrar una asignación adecuada a pesar de basarse en datos aproximados (tiempos de cómputo y comunicación de las tareas) a los valores reales de la aplicación [18] [19].

En este trabajo se estudia dicha característica en el algoritmo de asignación MATEHa desarrollado para el modelo TTIGHa. El estudio se realiza mediante un conjunto de pruebas que analizan el comportamiento del algoritmo frente a variaciones en los parámetros de entrada del mismo (tiempos de ejecución y de comunicación entre las tareas).

Trabajo Desarrollado

Como se mencionó en la sección anterior se realizaron diferentes pruebas experimentales para probar la robustez del algoritmo. Para la experimentación se realizaron los siguientes pasos:

1- Elegir la arquitectura para las pruebas.
2- Elegir el conjunto de aplicaciones con distintas características a evaluar.
3- Generar la asignación para cada una de las tareas de cada aplicación utilizando el algoritmo MATEHa.
4- Probar la robustez de cada una de estas asignaciones.

Elegir la arquitectura para las pruebas

La arquitectura heterogénea utilizada está compuesta por dos clusters conectados mediante un switch. El primero de ellos está formado por 20 procesadores cada uno de 2.4 Ghz Pentium IV con 1G RAM (denominado de acá en adelante como *cluster 1*). El segundo está compuesto por 10 procesadores, donde cada uno de ellos es de 2 GHz Celeron con 128 M RAM (llamado *cluster 2* de acá en más). La conexión entre los procesadores dentro de cada cluster es a través de una red Ethernet de 100 Mbits.

Esta arquitectura fue elegida de manera que los clusters que la componen sean de características diferentes en cuanto a la potencia de cálculo de los procesadores.

Para las pruebas se escogieron diferentes subconjuntos de procesadores de cada uno de los clusters, formando cuatro configuraciones:

- Configuración 1: 4 procesadores pertenecientes al cluster1.
- Configuración 2: 3 procesadores pertenecientes al cluster1 y 1 perteneciente al cluster2.
- Configuración 3: 2 procesadores pertenecientes al cluster1 y 2 pertenecientes al cluster2.
- Configuración 4: 1 procesador perteneciente al cluster1 y 3 pertenecientes al cluster2.

Elegir el conjunto de aplicaciones diferentes a evaluar

Se eligió un conjunto diferente de aplicaciones. Cada una de éstas variaba en cuanto a cantidad de tareas de la aplicación, tamaño de las tareas, cantidad de subtareas que componen una tarea y volumen de las comunicaciones entre subtareas. Todas estas características deben ser configuradas para cada aplicación.

En cada una de las pruebas realizadas para las diferentes aplicaciones se debe indicar primero la configuración de la arquitectura a utilizar. Una vez que se ha elegido la arquitectura debe especificarse los tipos diferentes de procesadores, la cantidad de procesadores para cada uno de estos tipos, los tipos diferentes de comunicación, el tiempo de startup y transferencia para cada uno de estos tipos, y por último el tipo de comunicación utilizado entre cada par de procesadores.

Una vez que esta información fue especificada se crea el grafo G generado a partir del modelo TTIGHa.

Generar la asignación para cada una de las tareas de cada aplicación utilizando el algoritmo MATEHa.

Utilizando el grafo generado para cada prueba en el punto anterior, se realiza la asignación mediante el algoritmo MATEHa y se lo compara con la asignación óptima (que se obtiene

analizando cómo funciona el algoritmo para todas las posibles asignaciones de tareas a procesadores). Esta comparación se realiza para poder determinar la eficacia de la asignación del algoritmo en cada una de las diferentes pruebas.

A modo de comentario en este punto se puede citar los resultados de un trabajo previo en el cual para las pruebas descriptas anteriormente el porcentaje de diferencia frente a la asignación óptima no supera el 12%.

Probar la robustez de cada una de estas asignaciones

La robustez de un algoritmo está relacionada con la sensibilidad de las variaciones en la estimación de los parámetros de entrada del modelo. Para el modelo utilizado, los parámetros que pueden no ser exactos al momento de calcular la asignación son: tiempo de ejecución de cada tarea en cada tipo diferente de procesador y los diferentes tiempos de comunicación dependientes de la red utilizada para la misma.

Dado que se utiliza un mapping estático, se presupone que las variaciones en los parámetros mencionados anteriormente son pequeñas y por lo tanto se estudia la influencia de las mismas en el algoritmo de asignación MATEHa.

Para concluir sobre el grado de robustez del algoritmo se analiza la sensibilidad del mismo con respecto al tiempo de ejecución de las tareas y a los diferentes tiempos en las comunicaciones.

Para analizar la sensibilidad del algoritmo MATEHa frente a posibles variaciones en sus parámetros de entrada se realizan una serie de pruebas incluyendo diferentes porcentajes de “ruido” (variaciones). Estas pruebas consideraron tres tipos de variaciones: sólo en los tiempos de cómputo de las tareas, sólo en los tiempos de las comunicaciones, y variaciones en los tiempos de cómputo de las tareas y en las comunicaciones de las mismas.

Experimentación Realizada

Para cada una de las diferentes aplicaciones definidas en el punto 2, las cuales consideraban distintas características en cuanto a las tareas que componían la aplicación y las comunicaciones, se realizaron pruebas agregando diferentes porcentajes de variaciones en el tiempo de cómputo y/o comunicación. Cada variación que se considera es un valor aleatorio entre 0 y un porcentaje máximo (diferente según cómputo o comunicación). Los valores para el porcentaje máximo tenidos en cuenta son del 0 al 100% en intervalos de 10%. Para obtener una muestra más significativa se generaron 10 corridas para cada una de estas variaciones.

En cada prueba se realizaron los siguientes pasos:

a) Para la aplicación a ejecutar se calcula el modelo TTIGHa.
b) Se obtiene la asignación (por medio del algoritmo de mapping MATEHa) para esa aplicación de acuerdo a los tiempos indicados en la prueba.
c) Se calculan los nuevos tiempos de cómputo y/o comunicación sumándole el porcentaje de variación correspondiente.
d) Con la asignación obtenida en b) y los nuevos tiempos calculados en c) se genera la simulación de la ejecución de la aplicación, para obtener el tiempo final.
e) Con los tiempos obtenidos en b) se calcula la asignación, también utilizando MATEHa y luego se realiza la simulación para dicha asignación.

f) Se comparan los tiempos finales obtenidos por las simulaciones de los puntos d) y e). Cuanto más cercanos son dichos tiempos implica que la asignación lograda por el algoritmo MATEHa se ve poco afectado por variaciones en los tiempos del modelo.

Resultados Obtenidos

Para poder analizar los resultados obtenidos, a partir de los mismos se calcula para cada uno de los distintos porcentajes (0..100%) de variación:

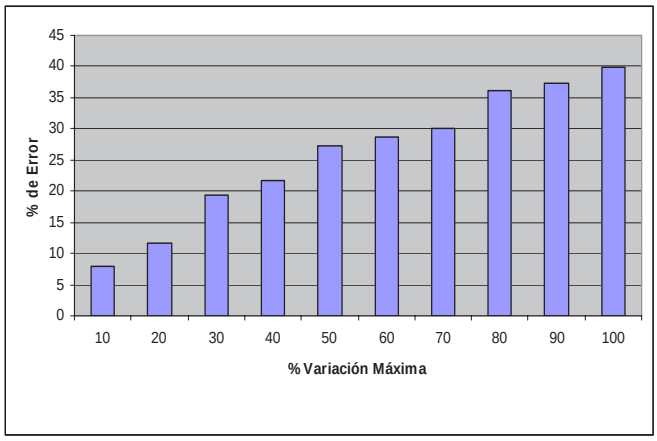
- Porcentaje de pruebas en las que hubo error, es decir, en las que el tiempo final obtenido en los puntos d) y e) fue diferente (% con Error).
- Error promedio. El error en una prueba está dado por la diferencia entre los tiempos obtenidos en d) y e) respecto al tiempo obtenido en 5 (Error Promedio General).
- Error promedio de las pruebas que tuvieron resultados diferentes en d) y e), este valor se calculó para realizar un análisis mas detallado de cuanto era la influencia del error en los resultados (Error Promedio Podado).

La Tabla 1 muestra algunos de los resultados obtenidos para las pruebas en donde las variaciones son en los tiempos de cómputo y comunicación. El conjunto completo de resultados se encuentra en [20]

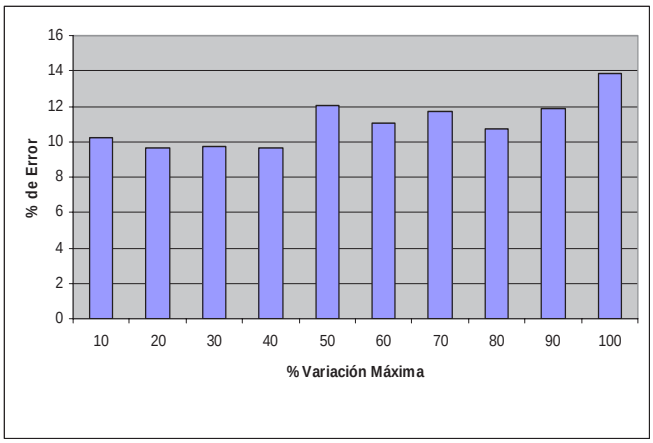
% de Variaciones (Cómputo-Comunicación)	% con Error	Error Prom. General	Error Prom. Podado
0-10	10,237	0,003	0,037
0-20	9,687	0,002	0,028
0-30	9,765	0,002	0,029
0-40	9,609	0,003	0,032
0-50	12,031	0,003	0,029
0-60	11,015	0,003	0,033
0-70	11,718	0,004	0,034
0-80	10,703	0,003	0,031
0-90	11,875	0,004	0,039
0-100	13,828	0,004	0,032
10-0	7,968	0,003	0,042
10-10	16,95	0,005	0,032
10-20	17,97	0,006	0,033
10-30	19,77	0,006	0,032
10-40	15,78	0,005	0,036
10-50	16,02	0,006	0,042
10-60	18,28	0,007	0,038
10-70	18,05	0,005	0,031
10-80	18,44	0,005	0,028
10-90	17,89	0,005	0,031
10-100	18,59	0,006	0,036
40-0	21,640	0,009	0,042
40-10	27,34	0,011	0,043
40-20	29,14	0,010	0,034
40-30	31,33	0,011	0,035
40-40	30,08	0,011	0,037
40-50	30,00	0,012	0,040
40-60	29,92	0,012	0,041
40-70	30,39	0,010	0,033
40-80	32,42	0,012	0,038
40-90	29,30	0,010	0,034
40-100	29,45	0,010	0,034
60-0	28,75	0,013	0,047
60-10	34,77	0,016	0,048
60-20	35,47	0,015	0,043
60-30	32,42	0,014	0,045
60-40	33,83	0,015	0,044
60-50	36,88	0,015	0,042
60-60	35,94	0,015	0,043
60-70	34,61	0,015	0,043
60-80	35,16	0,015	0,044
60-90	33,91	0,013	0,039
60-100	33,75	0,012	0,037
100-0	39,765	0,022	0,056
100-10	40,23	0,022	0,056
100-20	42,50	0,021	0,051
100-30	38,44	0,020	0,052
100-40	42,42	0,021	0,049
100-50	43,91	0,023	0,052
100-60	43,20	0,024	0,056
100-70	41,33	0,022	0,055
100-80	41,25	0,023	0,056
100-90	43,83	0,018	0,043
100-100	45,39	0,023	0,052

Tabla 1. Resultados de las pruebas con variaciones en los tiempos de cómputo y comunicación

La Figura 1a) muestra el porcentaje de error obtenido para los diferentes valores en las variaciones de cómputo, mientras que la Figura 1b) presenta el porcentaje de error obtenido para los diferentes valores en las variaciones de las comunicaciones.



a) Porcentaje de error con diferentes variaciones en el cómputo

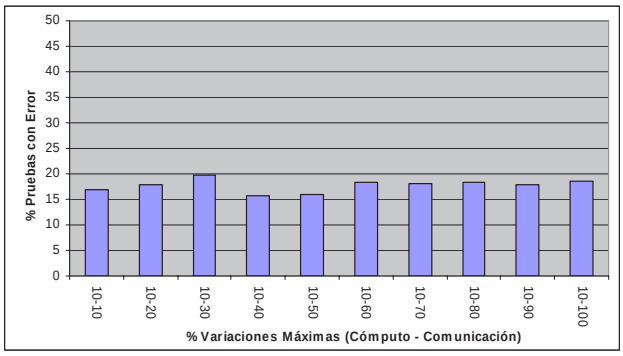


b) Porcentaje de error con diferentes variaciones en la comunicación

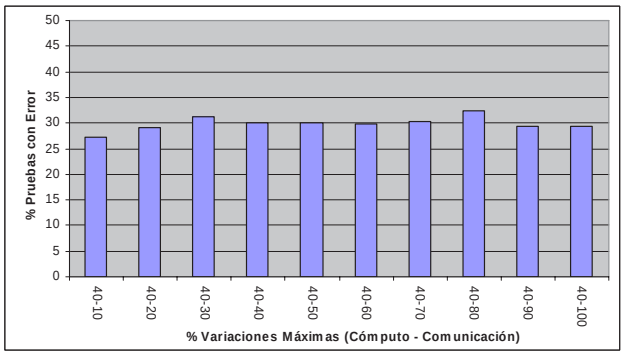
Figura 1

De las Figuras 1a) y 1b) se puede notar que al variar el porcentaje de cómputo esto genera que el error aumente, sin embargo no ocurre en la misma medida con el porcentaje de error cuando se varían los valores en las comunicaciones.

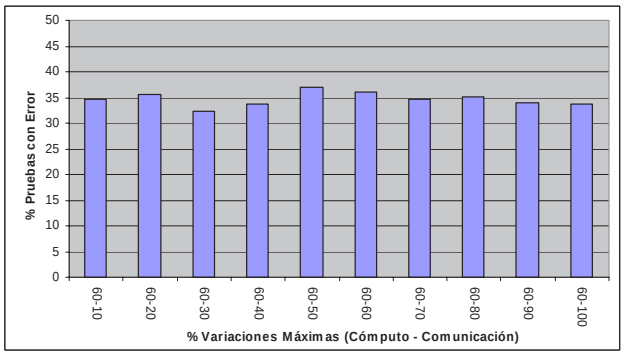
La Figura 2 muestra un conjunto de gráficos que consideran variaciones tanto en el cómputo como así también en las comunicaciones.



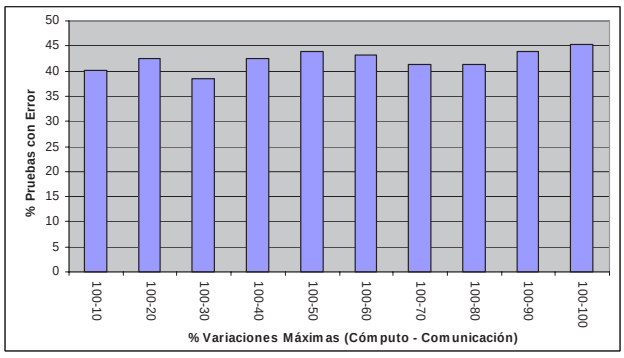
Porcentaje de pruebas con error con
10 % de variación en el cómputo.



Porcentaje de pruebas con error con
40 % de variación en el cómputo.



Porcentaje de pruebas con error con
60 % de variación en el cómputo.

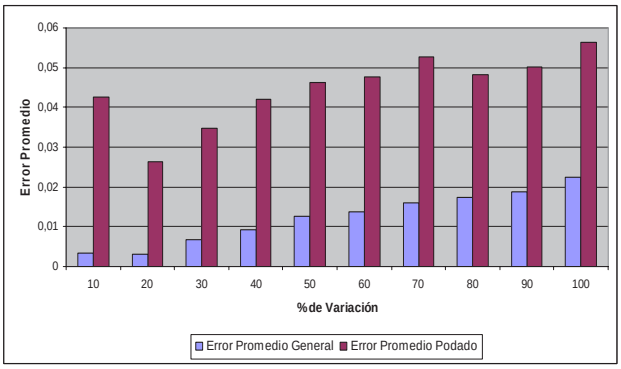


Porcentaje de pruebas con error con
100 % de variación en el cómputo.

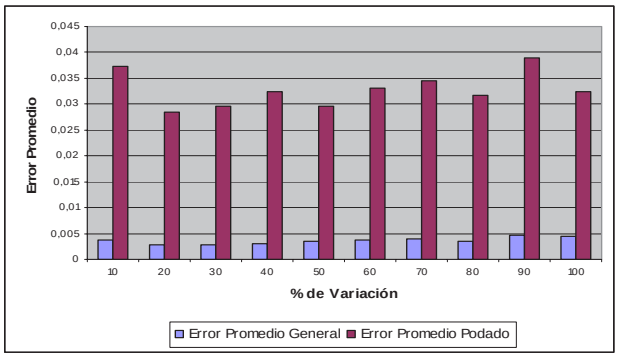
Figura 2

Al combinar las variaciones tanto en los tiempos de cómputo como en los de comunicaciones, se puede ver que respecto a los porcentajes de pruebas donde se detectó errores mantiene los lineamientos de cuando solo se analizó las variaciones en el cómputo, pero con un leve aumento.

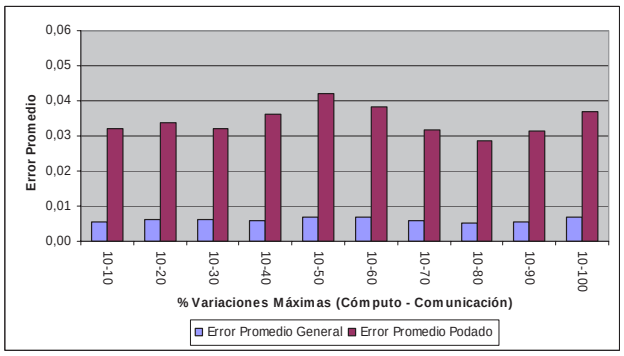
Como se describió anteriormente también se analizó el promedio de error cometido en las pruebas en las que hubo error. La Figura 3 permite observar estos promedios.



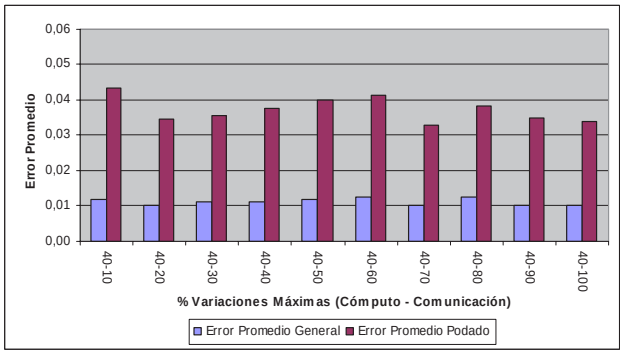
Porcentaje de error general y podado utilizando diferentes variaciones de cómputo



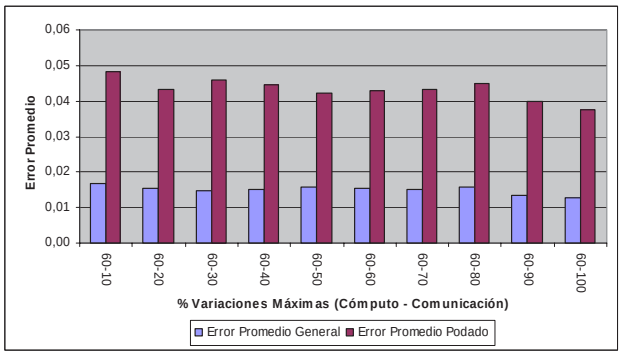
Porcentaje de error general y podado utilizando diferentes variaciones de comunicación



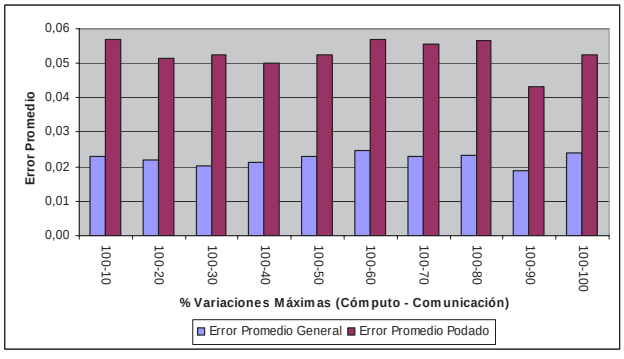
Error promedio general y podado con 10 % de variación en el cómputo.



Error promedio general y podado con 40 % de variación en el cómputo.



Error promedio general y podado con 60 % de variación en el cómputo.



Error promedio general y podado con 100 % de variación en el cómputo.

Figura 3

La Figura 3 permite observar el error promedio general y el podado para las combinaciones de porcentajes de variación evaluados. En cada gráfico se mantiene fijo el porcentaje de variación respecto al cómputo, variando el de comunicación. En las mismas se puede ver que para todas las

variaciones el porcentaje de error podado no supera el 6%, mientras que el porcentaje de error general no supera el 2.5%.

Al igual que con el porcentaje de pruebas con error, al combinar las variaciones tanto en los tiempos de cómputo como en los de comunicaciones, se observa que tanto el error promedio general como el podado mantienen la forma de cuando se analizaron sólo variaciones en el cómputo, es decir aumentan a medida que se incrementa el porcentaje de variación del tiempo de computo.

Conclusiones

La variación en el cómputo tiene influencia sobre la asignación a realizar por el algoritmo, ya que al aumentar dicha variación también aumenta el porcentaje de error. Sin embargo en las pruebas realizadas anteriormente se puede notar que utilizando una variación de hasta el 60% el porcentaje de pruebas con error no supera el 30%.

En las pruebas que se realizaron sólo con variaciones en los tiempos de las comunicaciones se puede observar que el porcentaje de error respecto del mapeo óptimo no supera el 14% aún haciendo variaciones del 100%. Además los gráficos también permiten ver que en general el error se mantiene prácticamente constante.

En las pruebas en las cuales se realizaron variaciones tanto en los tiempos de cómputo como en las comunicaciones se puede observar que, en cuanto al porcentaje de pruebas con error, se mantiene lo que ocurre al variar uno de los dos tipos de tiempo; resulta creciente de acuerdo a la variación en los tiempos de cómputo y se mantiene constante en relación a las variaciones de los tiempos de comunicación. Es decir que conserva la forma de cuando sólo se varía el tiempo de cómputo pero con un leve incremento relativamente constante por variar el tiempo de comunicación. En este caso, se llega a un 37 % de error al usar una variación de 60% en el cómputo.

Con respecto al error promedio podado, se observa un leve crecimiento a medida que aumenta la variación en el tiempo de cómputo, sin embargo en ningún caso supera el 6 %. Por ultimo, en el error promedio general ocurre lo mismo, sin superar el 2.5 %. Estos resultados permiten concluir que el algoritmo MATEHa presenta un alto grado de robustez, ya que logra realizar una buena asignación sin la necesidad de utilizar los parámetros exactos en cuanto a tiempo de cómputo y comunicación.

Bibliografía

[1] Grama A., Gupta A., Karypis G., Kumar V., "An Introduction to Parallel Computing. Design and Analysis of Algorithms", Pearson Addison Wesley, 2nd Edition, 2003
[2] Hagit Attiya, Jennifer Welch, Distributed Computing: Fundamentals, Simulations, and Advanced Topics (Wiley Series on Parallel and Distributed Computing). Wiley-Interscience; 2 edition (March 12, 2004).
[3] Kenneth A. Berman, Jerome L. Pau, Algorithms: Sequential, Parallel, and Distributed. Course Technology; 1 edition (October 11, 2004).
[4] Leopold C., "Parallel and Distributed Computing. A survey of Models, Paradigms, and Approaches", Wiley Series on Parallel and Distributed Computing. Albert Zomaya Series Editor, 2001
[5] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. Thee Design and Analysis of Computer Algorithms. Addison-Wesley, Reading, Massachusetts, 1974
[6] Akl S, “Parallel Computation. Models and Methods”, Prentice-Hall, Inc., 1997.

[7] Bohn C., Lamont G., "Load Balancing for Heterogeneous Clusters of PCs", Future Generation Computer Systems, Elsevier Science B.V., Vol 18, 2002, pp 389-400
[8] Goldman. “Scalable Algorithms for Complete Exchange on Multi-Cluster Networks”. CCGRID'02, IEEE/ACM, Berlin, p. 286 - 287, 2002.
[9] C. Roig, A. Ripoll, M.A. Senar, F. Guirado, and E. Luque. Modelling Message-Passing Programas for Static Mapping. In Euromicro Workshop on Parallel and Distributed Processing (PDP’00). IEEE CS Press. USA, pp 229-236, 1999
[10] J.J. Hwang, Y.C. Chow, F.D. Anger, and C.Y. Lee. Scheduling Precedence Graphs in Systems with Interprocessor Communication Times. SIAM Journal of Computing, 18(2): 244-257, April 1989
[11] A. Kalinov, S. Klimov. Optimal Mapping of a Parallel Application Processes onto Heterogeneous Platform. Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS’05), April 2005
[12] C. Roig, “Algoritmos de asignación basados en un nuevo modelo de representación de programas paralelos”, Tesis Doctoral, Universidad Autónoma de Barcelona, 2002.
[13] Laura De Giusti, Franco Chichizola, Marcelo Naiouf, Ana Ripio, Armando De Giusti, “ A Model for the Automatic Mapping of Task to Proccessors in Heterogeneous Multicluster Architecture”, Journal of Computer Science and Technology , Vol 7 nro 1 ISSN: 1666-6046, pag 39-44, Marzo 2007 .
[14] J. Cuenca, D. Gimenez, and J. Martinez, “Heuristics for Work Distribution of a Homogeneous Parallel Dynamic Programming Scheme on Heterogeneous Systems”, Procss of the 3rd International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks (HeteroPar’04), July 5-8, 2004 Cork, Ireland, IEEE CS Press.
[15] M. Garey and D. Johnson. Computers and Intractability. W.H. Freeman and Co. S. Francisco, 1979
[16] J.C. Cunha, P. Kacksuk, and S.C. Winter. Parallel Program development for cluster computing. Nova Science Pub. Inc., Huntington, New York, 2001
[17] C. Roig, A. Ripoll, M. Senar, F. Guirado, and E. Luque. “Exploiting knowledge of temporal behavior in parallel programs for improving distributed mapping”. 6th. International Euro-Par conference. Lecture Notes in Computer Science,1900:262–271, 2000.
[18] Ali, S., Maciejewski, A.A., Siegel, H.J., Kim, J.-K., 2003. Definition of a robustness metric for resource allocation. In: Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS). p. 42.
[19]Darin England, Jon Weissman, and Jayashree Sadagopan, A New Metric for Robustness with Application to Job Scheduling,
IEEE International Symposium on High Performance Distributed Computing 2005 (HPDC-14), July 24-27, 2005, Research Triangle Park, NC.
[20] Franco Chichizola, Laura De Giusti. Reporte Técnico: “Pruebas Realizadas para al algoritmo de mapping MATEHa asignando Variaciones de Cómputo y Comunicación a los parámetros del modelo TTIGHa”, Marzo 2007.

Balanceo Distribuido del Encaminamiento para el Control de Congestión en Redes Infiniband

Diego Lugones, Daniel Franco, Emilio Luque

Departament d'Arquitectura de Computadors i Sistemes Operatius,
Universitat Autònoma de Barcelona, 08193-Bellaterra, Barcelona, Spain
Ph. +34-3-5812888
diego.lugones@aomail.uab.es; daniel.franco@uab.es; emilio.luque@uab.es

Abstract

Communications requirements in High Performance Computing (HPC) demand the use of Interconnections Networks to connect processing nodes. Sharing resources in high performance interconnection networks leads to message congestion. Congestion spreading increases latency and reduces network throughput causing important performance degradation. Nowadays most current techniques use message throttling to prevent injection of new messages in network congested region. Message throttling moves contention from switches to sources nodes in order to eliminate congestion, however global latency is highly incremented because of the time that packets must wait in the source node. In this paper, we propose a congestion control mechanism for InfiniBand networks based in an adaptive routing algorithm that perform a communication load balancing over several alternative paths, in order to take load away of the congested network zone, eliminating congestion and maintaining injection rate. Our mechanism's experimentation results show latency, throughput and dynamic behaviour improvement over InfiniBand original congestion control mechanism which is based in message throttling. The proposed mechanism use the management model defined in InfiniBand specs, thus full compatibility is provided.

Keywords: Parallel processing, Communications and networks, Adaptive routing algorithms, Congestion control, InfiniBand networks, Hot-spot avoidance, High Speed Interconnection Networks, Network monitoring, Communication load balancing.

Resumen

El uso de recursos compartidos en las redes de interconexión de alta performance puede provocar situaciones de congestión de mensajes que degradan notablemente las prestaciones, aumentando la latencia de trasporte y disminuyendo la utilización de la red. Hasta el momento las técnicas que intentan solucionar este problema utilizan la regulación de la inyección de mensajes. Esta limitación de la inyección traslada la contención de mensajes desde los conmutadores hacia los nodos fuente, incrementando el valor de la latencia promedio global, pudiendo alcanzar valores muy elevados. En este artículo, proponemos una técnica de control de congestión para redes InfiniBand basada en un mecanismo de encaminamiento adaptativo que distribuye el volumen de comunicaciones entre diversas trayectorias alternativas quitando carga de la zona de congestión, lo que permite eliminarla. La experimentación realizada muestra la mejora obtenida en latencia y *throughput*, respecto al mecanismo de control de congestión original de InfiniBand basado en la regulación de la inyección. El mecanismo propuesto es totalmente compatible y no requiere que se modifique ningún aspecto de la especificación, debido a que se utilizan componentes de gestión definidos en el estándar InfiniBand.

Palabras clave: Procesamiento Paralelo, Comunicaciones y Redes, Algoritmos de encaminamiento adaptativos, Control de congestión, Redes InfiniBand, Evitación de Hot-spots, Redes de interconexión de alta velocidad, Monitorización de la red de interconexión, Balanceo de la carga de comunicaciones.

1 INTRODUCCION

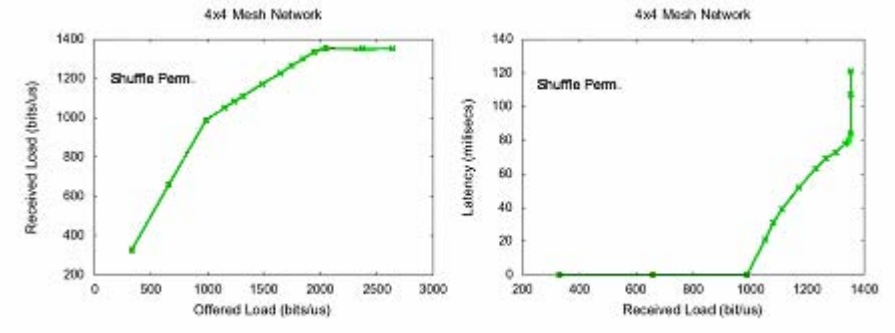
La evolución en el campo de las redes de interconexión para sistemas de cómputo de altas prestaciones *(High Performance Computing, HPC)* ha sido constante en los últimos años. Los avances tecnológicos han permitido una mejora importante en la velocidad de transmisión, incrementando significativamente el ancho de banda de los enlaces. Dichos avances también han tenido gran impacto en la integración de puertos en los conmutadores aumentando la cantidad de conexiones y permitiendo obtener topologías mas complejas y flexibles.

La reciente aparición de redes de interconexión comerciales como InfiniBand, Myrinet, Quadrics,…, etc. con alta velocidad de transmisión de datos, permiten construir las redes de interconexión que necesitan los sistemas de cómputo de altas prestaciones. Estas redes han tenido impacto no sólo en estos sistemas de cómputo, donde magnitudes adecuadas de velocidad de comunicación y tiempo de viaje de mensajes son de extrema importancia, sino también, en otros con menos requerimientos de performance, como redes de sistema (SAN) y clusters de ordenadores [12]. Los nuevos estándares desarrollados y las implementaciones comerciales deben su existencia a la creciente demanda de aplicaciones con grandes requisitos de cómputo.

El principal problema en el diseño de las redes de interconexión radica en manejo inadecuado de la congestión de mensajes en transito [3]. Dicha congestión aparece debido al uso compartido de los recursos de la red de interconexión (enlaces, buffers y conmutadores) y si esta situación no se controla eficientemente, es posible alcanzar la saturación de dichos recursos. Cuando la red no es capaz de manejar el volumen de comunicaciones que recibe en un momento dado, los mensajes en transito deberán competir por los recursos. Esta situación deriva en un aumento en el tiempo de viaje de los mensajes (latencia) y se propaga rápidamente a toda la red teniendo como efecto principal un deterioro global en la performance del sistema.

En la Figura 1 se ilustra la degradación de prestaciones en una red InfiniBand de 16 nodos y puede verse el comportamiento de la latencia de transporte de los mensajes y la carga de tráfico presente en la red. Cuando el volumen de mensajes inyectados presenta cargas bajas y medias de tráfico, la red es capaz de manejar este volumen ofrecido de mensajes y entregarlos a sus destinos; por lo tanto la carga aceptada por los nodos es igual a la carga inyectada (Figura 1 (a)) y la latencia se comporta de manera lineal y acotada (Figura 1 (b)). A medida que el tráfico aumenta y su valor alcanza un nivel determinado se observa que el tráfico recibido es menor al ofrecido, asimismo la latencia de los mensajes aumenta considerablemente. En este punto la red ingresa en el estado de saturación y se observa una degradación importante en sus prestaciones.

Las consecuencias derivadas de la congestión son aun mas graves en redes que no permiten el descarte de paquetes, como es el caso de las redes que conforman la mayoría de computadores paralelos de altas prestaciones [3].



(a) **(b)**

Figura 1. Degradación de performance en una red Infiniband con topología malla de 16 nodos. Patrón de trafico *Shuffle permutation*. Longitud de mensaje 4kB.

La solución tradicional al fenómeno de la congestión, radica pues en diseñar la red utilizando una cantidad de recursos mayor a la estrictamente necesaria *(Sobredimensionar la red)* de manera que no exista la necesidad de competir por dichos recursos y evitar la retención de paquetes [8]. Sin embargo esta practica ha quedado obsoleta debido, tanto al aumento en el coste de los componentes de red con respecto al de los procesadores empleados en los computadores paralelos actuales, como al elevado consumo de potencia de dichos componentes debido fundamentalmente al aumento de la velocidad de los enlaces [4]. Es por esta razón que el diseño de redes de interconexión sobredimensionadas no es una solución viable en la construcción de sistemas de cómputo reales.
La otra solución posible al problema de congestión, consiste en el empleo de técnicas *reactivas*, que a diferencia de las soluciones basadas en el sobredimensionamiento *(preventivas)*, utilizan los recursos estrictamente necesarios y los gestionan de forma eficiente. Estas técnicas monitorizan el estado de la red con el motivo de detectar la congestión y eliminarla mediante el uso de algún mecanismo.
Los mecanismos de control de congestión deben cumplir con un conjunto de características de diseño y operación que garanticen un uso eficiente de la red de interconexión y mejoren sus prestaciones de funcionamiento. En primer lugar el mecanismo de control de ofrecer un *tiempo de respuesta* convenientemente acotado para evitar reacciones tardías ante las situaciones de congestión que intentan controlar. Asimismo deben proveer un *balanceo global* de la carga de tráfico, ya que algunos mecanismos pueden proveer una respuesta adecuada a nivel local, pero conllevan un desbalance de carga en la red que conduce a una degradación global de las prestaciones. Por otra parte, ciertas técnicas reaccionan correctamente ante situaciones de congestión, pero *penalizan* el correcto funcionamiento de la red, debido a que generan sobrecarga de mensajes o nuevos fenómenos indeseables (ej. *deadlock, livelock, starvation* [8]) con cargas normales de trafico. El éxito de algunos mecanismos depende en gran parte de las condiciones de tráfico presente en la red, la topología ó el tamaño de los mensajes. Por esta razón el mecanismo de control de congestión debe ser *robusto* y mantener sus prestaciones en una amplia variedad de casos. Por ultimo la técnica utilizada debe ser *eficiente* y *escalable*, con el fin de eliminar el problema completamente y en una amplia variedad de topologías.
En este artículo presentamos el trabajo realizado en el diseño y la aplicación de un algoritmo de encaminamiento (denominado "*Distributed Routing Balancing*", DRB) vigente, factible y realista, citado recientemente en otros artículos y contribuciones, por distintos autores en estudios similares [1] y [4], sobre el emergente estándar InfiniBand cuya utilización esta ganando terreno velozmente en el campo de las redes de interconexión para computadores de alta performance, pero que carece de un control de congestión adecuado. En este sentido, mostraremos la experimentación realizada sobre nuestra propuesta de control de congestión, basada en el balanceo de la carga de comunicaciones y la importante mejora en las prestaciones con respecto al mecanismo de control original que ofrece InfiniBand y que se basa en la regulación de la velocidad de inyección de los mensajes.
El resto del artículo esta organizado de la siguiente manera: En la sección 2 se describen los antecedentes en los mecanismos de control de congestión y la descripción de DRB. La sección 3 presenta las características principales de la arquitectura InfiniBand que hacen posible la aplicación del balanceo distribuido del encaminamiento. En la sección 4 se describe nuestra propuesta de control de congestión. Los resultados de la evaluación de las prestaciones ofrecidas por el mecanismo propuesto para diversas topologías y distribuciones de tráfico, se presentan en la sección 5. Finalmente se presentan las conclusiones extraídas del trabajo realizado.

2 ANTECEDENTES

La capacidad de gestionar un alto número de mensajes sin que se produzca un gran aumento de la latencia es fundamental en las redes de interconexión de altas prestaciones, más aún cuando éstos

soportan aplicaciones en las que la relación de cómputo frente a comunicación es pequeña *(granularidad fina)*, y que por tanto generan un gran tráfico de mensajes entre los nodos de la red. Por este motivo, el control de congestión ha sido objeto de estudio durante los últimos años.
Las técnicas que intentan manejar y solucionar los problemas asociados a la congestión se basan en realizar una *monitorización* del tráfico que circula por la red, o de los recursos que la componen, con el motivo de *detectar* la congestión, *notificar* su existencia y llevar a cabo algún mecanismo para *controlarla y eliminarla*.
Algunas técnicas analizan el tiempo de bloqueo de mensajes (Latencia) [5], o el nivel de recursos ocupados (Canales o *Buffers*) [1][3][4] para determinar la existencia de congestión e informar de la misma al resto (o parte) de los nodos presentes en la red para que ejecuten las acciones que permiten evitar la degradación de performance en la red de interconexión.
Una de las acciones correctivas utilizadas por las técnicas de control, con el fin de eliminar la congestión es la típica regulación de la velocidad de inyección en el nodo fuente *(Message throttling)* [4][9][10] que detiene la inyección de nuevos mensajes, permitiendo que los conmutadores encaminen los mensajes ubicados en la zona congestionada hacia el destino manteniendo acotada la ocupación de los *buffers*. Esta técnica tiene efectos adversos que deterioran las prestaciones de la red, debido a que la latencia de transporte se incrementa considerablemente en virtud de la espera a que se someten los mensajes antes de su inyección.
Otra posibilidad utilizada en la eliminación de la congestión, consiste en el manejo y optimización del uso de los *buffers* en los puertos de los conmutadores *(switchs)* [1][3][4]. Estas soluciones son simples y fáciles de implementar, sin embargo no presentan buenas prestaciones debido principalmente a que no resuelven el problema de congestión a nivel global, sino que intentan manejar el volumen de comunicación mediante la organización de paquetes de forma local.
Por último, las técnicas basadas en el uso de encaminamiento adaptativo *(Adaptive routing)* [1][2][5] también permiten eliminar la congestión debido a que los mensajes son enviados a los destinos correspondientes, teniendo en cuenta el estado de los diferentes caminos posibles. De esta manera si algún puerto de un encaminador que pertenece a una determinada trayectoria se encuentra congestionado, el algoritmo modifica el envío utilizando trayectorias alternativas. Estas técnicas presentan mejores prestaciones pues permiten mantener el nivel de inyección de mensajes y actúan redistribuyendo la carga.
Recientemente, se han propuesto varias técnicas para el control de congestión en redes InfiniBand que intentan mejorar la utilización de la red. Estas técnicas [6][9][10] proponen la regulación de inyección de mensajes como mecanismo para eliminar la congestión. La principal desventaja de este mecanismo, radica en que la congestión se elimina trasladándola congestión desde los conmutadores hacia los nodos fuentes que inyectan los paquetes. De esta forma el comportamiento global de la latencia promedio se incrementa igualmente, pudiendo alcanzar valores muy elevados en presencia de cargas de tráfico adversas. Por otro lado la técnica propuesta en [2][13] permite el uso de trayectorias múltiples pero sólo puede utilizarse con el modelo de comunicaciones de conexión confiable *(Reliable Connection)* y no en los otros tres modelos soportados por Infiniband [1], ya que en estos casos no es posible realizar la detección y la notificación de congestión.

2.1 Balanceo distribuido del encaminamiento (DRB).

La técnica de control de congestión propuesta en este artículo está basada en un mecanismo de balanceo del encaminamiento que intenta uniformizar la carga en todos los enlaces de la red de interconexión. Este mecanismo se conocido como: Balanceo Distribuido del Encaminamiento ("*Distributed Routing Balancing*", DRB por sus siglas en inglés) [5] y se basa en la distribución uniforme de la carga en la red mediante la expansión de caminos. Esta expansión es dinámica y está controlada por el nivel de latencia existente en la red. El método establece nuevos caminos alternativos simultáneos entre cada par fuente y destino con objeto de mantener una latencia baja de

los mensajes. DRB define cómo crear los caminos alternativos para expandir los caminos simples originales y cuándo y cómo usarlos dependiendo del nivel de carga de tráfico en la red de interconexión. Es importante destacar que el mecanismo produce un efecto de balance colectivo (a nivel global), pues esta expansión se produce para todos los pares fuente-destino de la aplicación que interaccionan entre sí.
Conceptualmente, este método mide el estado de la carga en todas las conexiones entre pares fuente destino, al detectar congestión se notifica al/los nodo/s que inyectan mensajes para que configuren nuevas trayectorias posibles y redistribuyan el trafico según su estado de carga. Este funcionamiento se muestra en la figura 2, donde se observa que se detecta la presencia de congestión en los nodos intermedios y se notifica al nodo fuente su existencia mediante un mensaje de reconocimiento. A continuación este nodo determina las trayectorias alternativas que utilizara en el encaminamiento posterior de mensajes. Es importante destacar el concepto en el que el algoritmo basa su funcionamiento, porque puede ser implementado de diversas maneras. Es decir: la detección puede realizarse mediante la medición de la latencia acumulada a través del enlace, o en función de la ocupación de los buffers en los puertos de los conmutadores, etcétera. Mientras que la redistribución de tráfico puede llevarse a cabo de manera aleatoria, o en función de la ocupación de los enlaces, o de la velocidad de éstos en redes no homogéneas.

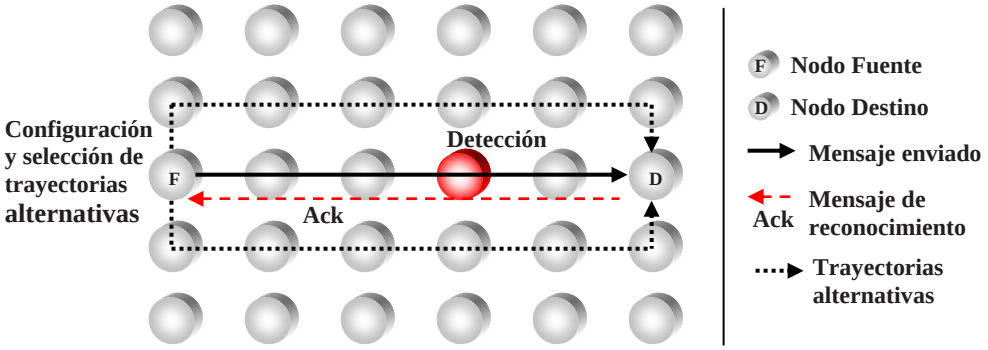


Figura 2. Fases de balanceo distribuido del encaminamiento DRB

Esta versatilidad en la implementación del algoritmo, junto a su buen comportamiento hace que su aplicación a tecnologías de interconexión actuales sea un tema de investigación interesante, tanto para la mejora y evaluación de esta técnica, como para desarrollo de nuevas propuestas sobre las tecnologías utilizadas.

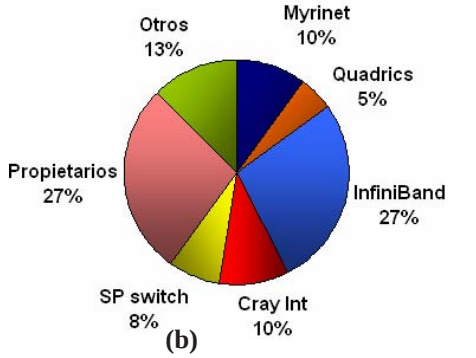
3 LA ARQUITECTURA INFINIBAND (IBA)

En la última década han aparecido una importante cantidad de especificaciones que acompañadas del avance tecnológico adecuado, pretenden desarrollar y estandarizar redes de comunicaciones con las características y requerimientos (conexiones punto a punto, baja latencia, elevado ancho de banda, etc.) necesarios en las redes de interconexión. Puntualmente, la especificación InfiniBand [1] es una nueva y poderosa arquitectura diseñada, no sólo para cubrir con las demandas de performance asociadas con el movimiento de datos en los dispositivos entrada-salida, sino también para conformar los cluster de cómputo de altas prestaciones *(High Performance Computing HPC)*, debido al elevado ancho de banda y la baja latencia de transporte que ofrece.
Los clusters InfiniBand de gran escala están ganado gran popularidad según lo reflejan los rankings de supercomputadores en el *top500* [11] tal como se muestra en la figura 3, donde puede verse como ha evolucionado InfiniBand en el último año, hasta alcanzar la segunda posición entre los sistemas de interconexión usados en los supercomputadores más potentes del mundo (figura 3(a)). La figura 3(b) muestra solamente los cuarenta primeros puestos, y se observa que InfiniBand es el estándar mas utilizado debido a las prestaciones que ofrece. Al mismo tiempo, las topologías directas (mallas, toros, hipercubos…) y la topología fat-tree se han convertido en las más utilizadas

en la interconexión para estos clusters, debido a que permiten múltiples trayectorias disponibles entre un mismo par de nodos. No obstante, incluso en estas topologías, pueden ocurrir situaciones de congestión, que dependen principalmente de la configuración de trayectorias entre nodos y del patrón de comunicación de la aplicación. Para empeorar aun más la situación, la naturaleza determinista del encaminamiento utilizado por defecto en InfiniBand, limita a las aplicaciones del uso eficaz y transparente de trayectorias múltiples que permiten evitar situaciones de congestión.

Interconexión	Junio 2007	Noviembre 2006	Junio 2006
Gigabit Ethernet	41.20 %	42.60 %	51.20 %
Myrinet	9.20 %	15.80 %	17.40 %
InfiniBand	25.60 %	15.60 %	7.20 %
Quadrics	2.20 %	2.80 %	2.80 %

(a)



(b)

Figura 3. Evolución y uso de IBA en el *top500*

InfiniBand define una arquitectura de red que permite interconectar múltiples nodos de procesamiento y dispositivos de entrada-salida, utilizando una red arbitraria con conexiones punto a punto como se muestra en la Figura 4. Los nodos de procesamiento pueden incluir varias CPUs y módulos de memoria, y utilizan uno o varios adaptadores de canal *(Channel Adapters, CAs)* como interfase para conectarse con los conmutadores de la red. La red se estructura en diferentes subredes *(subnets)* que interconectan los nodos de procesamiento a través de varios conmutadores *(switchs)*, las subredes se interconectan entre si mediante el uso de encaminadores *(routers)*.

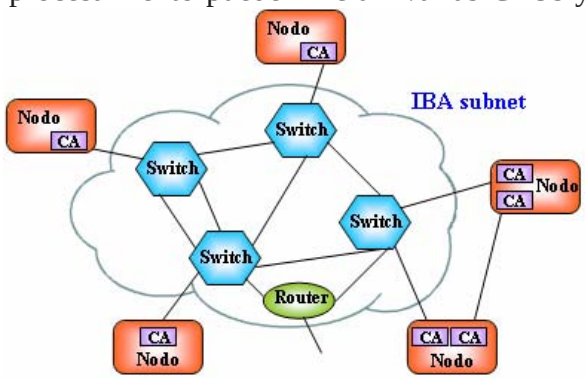


Figura 4. Subred InfiniBand

InfiniBand especifica un protocolo que divide la arquitectura en múltiples capas independientes, la capa física, la capa de enlace, la capa de red, la capa de transporte y las capas superiores. La capa de enlace y la de transporte conforman el corazón de la arquitectura, ya que en estas se crean los paquetes, se establecen las conexiones punto a punto y se realiza la conmutación dentro de la subred. IBA utiliza un mecanismo de canales virtuales (VL) para crear múltiples enlaces virtuales dentro de uno físico, mejorando su utilización.

Dentro de una subred, todos los dispositivos IBA tienen un identificador local de 16 bits *(Local Identificator, LID)* asignado por el gestor de subred *(Subnet Manager, SM)*. Los paquetes tienen incluido este identificador en sus cabeceras y los conmutadores lo utilizan para determinar el puerto de salida correspondiente y realizar el encaminamiento en la subred.

En IBA, las subredes se gestionan de una manera independiente usando un modelo de gestión *(Management Model)* en el que varias entidades se comunican para configurar el funcionamiento y las operaciones de la red. El gestor de subred (SM) es el encargado de descubrir los componentes de la subred y configurarlos (asignándoles los LIDs), y de activar y monitorizar la subred. El SM puede estar presente, tanto en un nodo de procesamiento como en un conmutador y puede implementarse tanto en software como en hardware. Las operaciones mencionadas se realizan utilizando paquetes de gestión que transportan la información entre el gestor de subred *(SM)* y los agentes de gestión *(Subnet Management Agent, SMA)* presentes en todos los dispositivos.

El gestor de subred busca los componentes, les asigna un identificador local y encuentra las trayectorias entre ellos. Asimismo puede configurarse para establecer varias trayectorias entre los

diferentes pares de nodo fuente-destino de la red utilizando una mascara de control *(Local Mask Control, LMC)*, que permite asignar hasta 2^{LMC} identificadores locales a cada puerto. Esta multiplicidad de nombres proporciona la manera de establecer múltiples trayectorias, entre el mismo par de nodos.

Otro componente importante del modelo de gestión es el gestor de control de congestión *(Congestion Control Manager, CCM)*, que provee los mecanismos necesarios para realizar la detección de congestión, generar y enviar notificaciones a los nodos fuente para que lleven a cabo la regulación de la inyección de mensajes.

Mediante el uso de elementos de medición, cada conmutador conoce el estado de ocupación en los buffers de cada canal virtual. La especificación establece un valor umbral de ocupación de los buffers, por encima del cual se activan los mecanismos destinados al control de congestión. Cuando se detecta congestión, el conmutador informa de esta situación marcando los paquetes que están situados en el buffer del canal virtual que ha superado el umbral. Dentro de la cabecera de transporte *(transport header)* de todos los paquetes, existe un bit destinado a tal efecto. Este bit es denominado *Forward Explicit Congestión Notification (FECN).* Una vez que el paquete es marcado, se reenvía por el puerto correspondiente en función la trayectoria especificada.

Si el nodo destino recibe un paquete marcado, el agente de control de congestión de dicho nodo solicita el envío de un mensaje de notificación *(congestión Notification, CN)*, con el objetivo de informar al nodo fuente la existencia de congestión en la trayectoria establecida entre ambos nodos. La notificación se hace efectiva mediante el uso de otro bit en la cabecera de transporte del mensaje CN, conocido como *Backward Explicit congestión Notification (BECN)*. El mensaje de notificación se envía hacia el nodo fuente. El agente de control de congestión analiza el bit *BECN* y responde informando al nodo que disminuya la inyección de mensajes. De esta manera, los puertos congestionados pueden recuperarse liberando los paquetes contenidos en sus buffers. La disminución en la inyección será más restrictiva dependiendo de la cantidad de mensajes de notificación recibidos.

Eventualmente, la congestión desaparece y la inyección debe recuperarse. Esta tarea se realiza utilizando un temporizador. Cada vez que transcurre un intervalo de tiempo sin que se hayan recibido mensajes de notificación, se recupera el nivel normal de inyección. Como se ha mencionado previamente, la principal desventaja de este mecanismo, radica en que la congestión se elimina trasladando la congestión desde los conmutadores, hacia los nodos fuentes que inyectan los paquetes. De esta forma el comportamiento global de la latencia promedio se incrementa igualmente, pudiendo alcanzar valores muy elevados en presencia de cargas de tráfico adversas.

Teniendo en cuentas estas razones, proponemos la aplicación del balanceo distribuido del encaminamiento en redes InfiniBand, utilizando en forma conjunta las características de establecimiento de trayectorias múltiples (ofrecido por el SM) y los mecanismos de monitorización de recursos y notificación mediante mensajes de reconocimiento (ofrecidos por el CCM), para aplicar un mecanismo de control de congestión eficiente, utilizando un concepto que permite un alto grado de utilización de los enlaces y un bajo valor de latencia de transporte en los mensajes, debido al uso de trayectorias alternativas que permiten la distribución de la carga de trafico, y no a la regulación de inyección propuesta en el mecanismo original.

4 BALANCEO DISTRIBUIDO DEL ENCAMINAMIENTO EN REDES INFINIBAND

La aplicación del balanceo distribuido del encaminamiento se lleva a cabo dentro del contexto establecido por la arquitectura de InfiniBand, teniendo en cuenta las definiciones descriptas en la especificación y utilizando sus características. De esta manera no se requiere modificación alguna y se mantiene la compatibilidad con el estándar IBA.

La *detección* de la congestión se lleva acabo en los canales virtuales de cada puerto, monitorizando la ocupación de los mismos con respecto a un umbral relativo al tamaño del buffer. El valor del umbral es establecido por el CCM y los diversos valores posibles, varían entre los números 0 y 15; donde el valor 0 indica que ningún paquete ha de marcarse en este puerto, y el valor 15 especifica un umbral muy restrictivo, ver Figura 5. Cuando el mecanismo detecta congestión dentro del conmutador, informa al nodo destino poniendo a *uno* el bit denominado FECN dentro de la cabecera de transporte *(Base Transport Header, BTH)* presente en todos los paquetes del protocolo. A continuación el mensaje se transmite hacia el nodo destino, según la trayectoria especificada inicialmente.

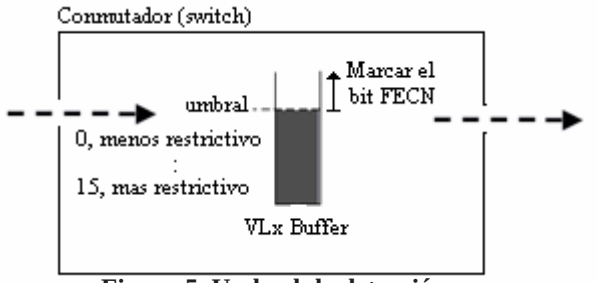


Figura 5. Umbral de detección

La *notificación* de la congestión se realiza una vez que el paquete con el bit de notificación FECN marcado alcanza el adaptador de canal *(CA)* del nodo destino, donde el agente de control de congestión *(CCA)* determina el envío de un mensaje de respuesta al nodo fuente, mediante un paquete de notificación de congestión, como se muestra en la Figura 6. En la cabecera de este paquete, existe un bit denominado BECN que informa explícitamente esta situación. Los nodos fuente que inyectan paquetes en zonas congestionadas de la red de interconexión comenzarán eventualmente a recibir notificaciones desde los nodos destino. En este momento, el agente de control de congestión (CCA) del nodo fuente activa el mecanismo que *configura y selecciona* los caminos alternativos. La apertura de las trayectorias se hace de forma gradual en función de la distribución de notificaciones recibidas (tal que las trayectorias alternativas son seleccionadas de forma inversamente proporcional a dicha distribución, es decir, los caminos menos ocupados son los más utilizados.) y las trayectorias son seleccionadas de manera que su largo no involucre un tiempo de transmisión demasiado grande, por lo que deben ser lo mas cortas posible. De esta manera, se consigue que el problema producido por la contención en los buffers no se traslade hacia una perdida de prestaciones provocada por un tiempo de viaje elevado.

Figura 6. Notificación

Para que la selección de trayectorias múltiples sea posible, el SM asigna un identificador local *(LID)* a cada puerto de la red, como se ha mencionado en la sección 3. El formato de este identificador, situado en las cabeceras de los paquetes, puede verse en la Figura 7. Cuando se reciben los paquetes dentro de un conmutador, los 8 bits menos significativos *(LMC)* son ignorados, de esta manera es posible modificar el valor de esta máscara para asignar a los puertos de los CA varios LIDs (es decir, varios nombres). De esta manera el SM puede establecer varias trayectorias para el mismo nodo, en virtud de la multiplicidad de nombres.

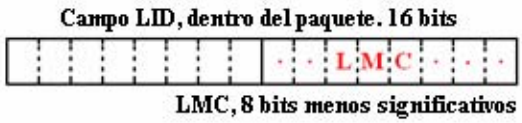


Figura 7. LID y LMC

En la fase inicial de configuración de la red, una vez que se recolecta toda la información de la misma, el gestor de subred entra en la fase de construcción de trayectorias. La especificación no define un algoritmo para cumplir con esta finalidad, por este motivo se ha implementado un mecanismo que genera caminos múltiples y selecciona los disjuntos, para cada par de nodos de la red. El mecanismo utilizado es el típico algoritmo de búsqueda en profundidad *(Depth-first search, DFS)*, con el agregado de una función que selecciona los caminos disjuntos. De esta manera, se configuran los caminos alternativos que serán utilizados cuando aparezca la congestión.

En ausencia de congestión, las trayectorias utilizadas para encaminar los paquetes dentro de la red de interconexión deben ser de largo mínimo con el objeto de ofrecer una baja latencia de transporte,

ya que en este caso, el valor de latencia esta principalmente determinado por la velocidad de los enlaces. La arquitectura especifica que cada agente de control de congestión debe contener un contador que permite medir el tiempo transcurrido desde que el último paquete de notificación ha arribado al nodo. La duración en la que este contador expira, es un parámetro que se establece mediante el gestor de control de congestión. El mecanismo propuesto en este artículo utiliza este contador para contraer las trayectorias en ausencia de congestión. Cada vez que el contador expira y no hayan llegado al nodo destino paquetes de notificación, el camino conformado por las múltiples trayectorias disjuntas se *contrae* gradualmente hasta recuperar la trayectoria original.
Mediante el uso de estas técnicas, se proporciona a la arquitectura InfiniBand, un mecanismo de control de congestión que mejora notablemente los resultados derivados del empleo del mecanismo especificado en el estándar, como se muestra en la sección siguiente.

5 EXPERIMENTACION Y EVALUACIÓN

En esta sección, se evalúa el comportamiento del mecanismo propuesto DRB y su mejora sobre la técnica de control de congestión que ofrece InfiniBand. El modelado de la arquitectura InfiniBand y las técnicas bajo estudio se realizan utilizando la herramienta de simulación estándar OPNET Modeler [8]. Esta herramienta provee un simulador DES *(Discrete Event Simulator)* [8], y ofrece un entorno de modelado jerárquico con técnicas de programación orientada a objetos que permite determinar el comportamiento de los componentes de red en base a la creación de maquinas de estados finitos *(Finite State Models)*. Las métricas más significativas en el estudio de las prestaciones en las redes de interconexión son: la latencia de transporte, que representa el tiempo requerido para entregar un mensaje desde su generación, incluyendo el tiempo en que se almacena en el buffer de salida del nodo fuente y el rendimiento *(throughput)* que representa el tráfico máximo aceptado por la red. Este tráfico se mide en bits/μs y la latencia en milisegundos. Las métricas descriptas presentan el comportamiento global de la red en valores promedio. Por esta razón es también necesario el estudio de la respuesta temporal de los mecanismos analizados con el fin de evaluar parámetros dinámicos como el tiempo de respuesta, la sobrecarga de mensajes en la red, etcétera. Por este motivo, también se ha evaluado la distribución de carga en los enlaces para comprobar las mejoras conseguidas con DRB sobre las características dinámicas de funcionamiento de la red. Los modelos utilizados en la simulación de los nodos de procesamiento y los conmutadores describen el funcionamiento de las capas del protocolo a nivel físico, de enlace y de transporte. También se han modelado los elementos de gestión de subred y de control de congestión que permiten la aplicación de *DRB*. Cada nodo contiene un adaptador de canal, y los conmutadores contienen varios puertos físicos con tres canales virtuales cada uno y el crossbar que permite la interconexión de los puertos de entrada con los puertos de salida. La metodología de evaluación desarrolla varios puntos. El primero consiste en evaluar DRB para un conjunto de redes de interconexión de diversos tamaños (toros y mallas) y de patrones de comunicación. La experimentación se realiza de forma exhaustiva y se enfoca en la respuesta en latencia y *throughput* a patrones de comunicación persistentes tomados de aplicaciones numéricas (*"Butterfly"*, *"Perfect Shuffle"* y *"Matrix Transpose"*) [8]. El segundo punto, consiste en la evaluación de la respuesta de la red de interconexión respecto a un patrón de comunicaciones que provoca la aparición de un "*hot-spot*" con el que se evalúa la respuesta dinámica de la red a través de la utilización de los enlaces congestionados. Debido a la similitud en los resultados obtenidos en todos los casos y por razones de tamaño, solo se presenta un subconjunto representativo de resultados. En la Figura 8 se muestran las prestaciones obtenidas en una red InfiniBand de 64 nodos conectados en una topología toro bidimensional. Los resultados son similares para todos los patrones utilizados. DRB ofrece mejores resultados que la técnica de control IBA y se observa que la diferencia entre las curvas para cada patrón se incrementa a medida que lo hace la carga de tráfico de la red. Se puede observar que a cargas bajas (un ancho de banda menor que 400 bits/μs), las técnicas propuestas se comportan de

forma similar. Esto es importante porque implica que DRB no cambia el comportamiento de la red cuando no es necesario, de manera que no introduce ninguna sobrecarga *(overhead)* con demandas bajas de inyección.

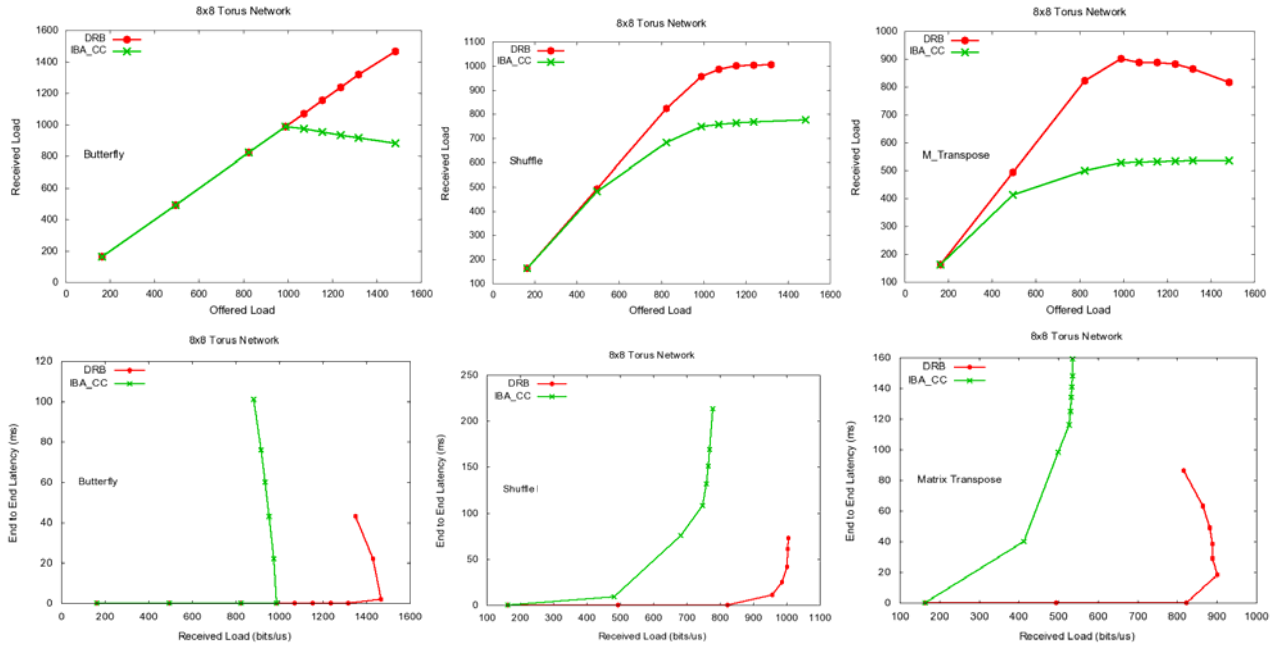


Figura 8. Prestaciones en una red Infiniband de 64 nodos con topología toro. *Throughput y latencia*

Cuando la inyección de trafico en la red se incrementa, con cargas entre 400 y 800 bits/µs, el incremento de la latencia obtenido con DRB es notablemente inferior al que se obtiene con la técnica InfiniBand, esto es debido a que, en DRB, los nodos fuente comienzan a utilizar caminos alternativos para el envío de paquetes, mientras que con IBA_CC los paquetes esperan en el nodo fuente. Con cargas elevadas en la red, para valores mayores a 800 bits/µs en la inyección, DRB utiliza el mayor número de caminos alternativos permitidos en la configuración (en este caso cuatro), resultando en valores de latencia menores respecto a la técnica InfiniBand. Al mismo tiempo que estas latencias se reducen, el *throughput* conseguido se mejora y se observa un incremento notable en la utilización de la red. Este aspecto puede verse en las graficas presentadas en primera fila de la Figura 8 donde se muestra la carga aceptada como función de la carga aplicada. La curva correspondiente a DRB representa una carga mayor que la correspondiente a IBA_CC, donde la red se satura antes y, por tanto, otorga valores más bajos de carga aceptada. Las ganancias con respecto al método original están entre el 35 y el 50%, según el patrón utilizado.

En la Figura 9 se muestran los resultados obtenidos para una red de 32 nodos conectados en una topología malla (o grid). Esta topología, por sus características físicas ofrece menor cantidad de caminos alternativos y por tanto el *throughput* producido es ligeramente peor que en el caso del toro.

Al igual que en el caso anterior, el comportamiento de DRB y el de IBA_CC son similares con cargas muy bajas, pero DRB es mejor en la zona de carga máxima, en la que presenta menores latencias y ofrece mejores prestaciones. Esto significa que DRB es capaz de soportar cargas mayores y se demuestra que, ante condiciones extremas, ofrece mejores prestaciones que el otro método debido a la distribución de caminos utilizada. Esto puede observarse en las dos topologías estudiadas donde el balanceo distribuido del encaminamiento ofrece ganancias entre 200 y 300% en la parte plana de la curva de latencia.

Estas ganancias se deben principalmente a que DRB aprovecha las trayectorias alternativas para el envío de mensajes, mientras que con la técnica IBA_CC deben esperar en el nodo fuente.

Según se ha mencionado anteriormente, hemos diseñado un experimento donde el patrón de comunicación definido provoca la aparición del "*hot-spot*", donde varios mensajes compiten por los

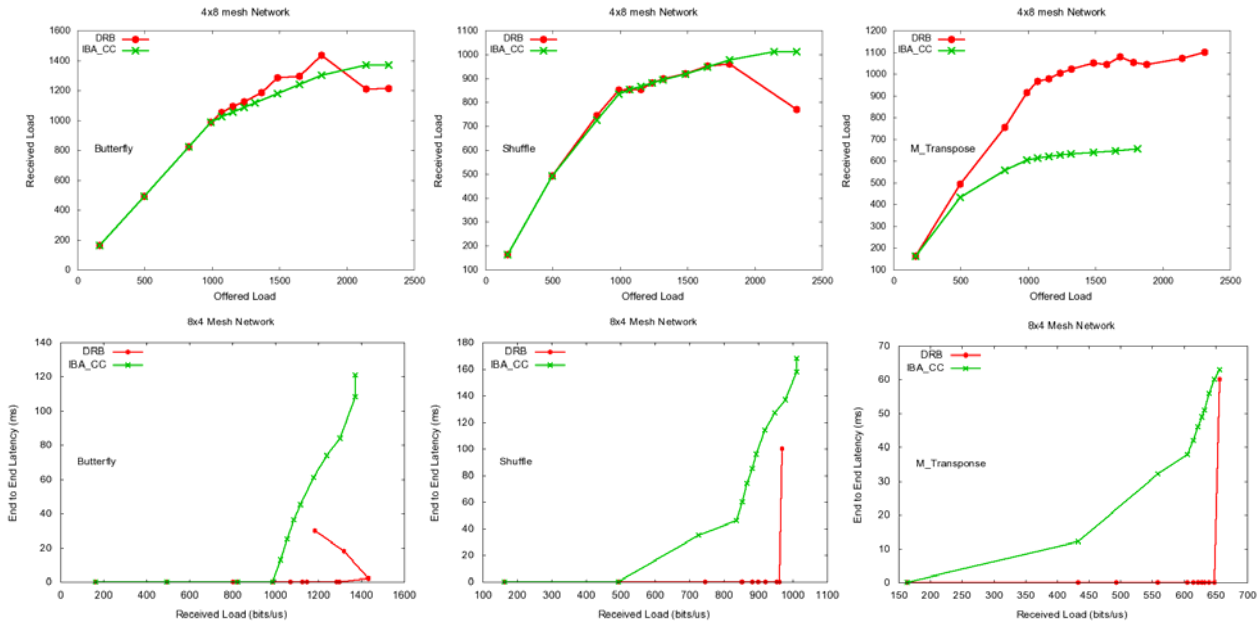


Figura 9. Prestaciones en una red Infiniband de 32 nodos con topología malla. *Throughput y latencia*

recursos sobre un camino común. Este patrón permite analizar y comparar las dos técnicas bajo condiciones extremas de carga aplicada. En la Figura 10 puede observarse el tráfico presente en los enlaces de la red, donde se genera repentinamente una gran carga localizada en una zona concreta de la red.

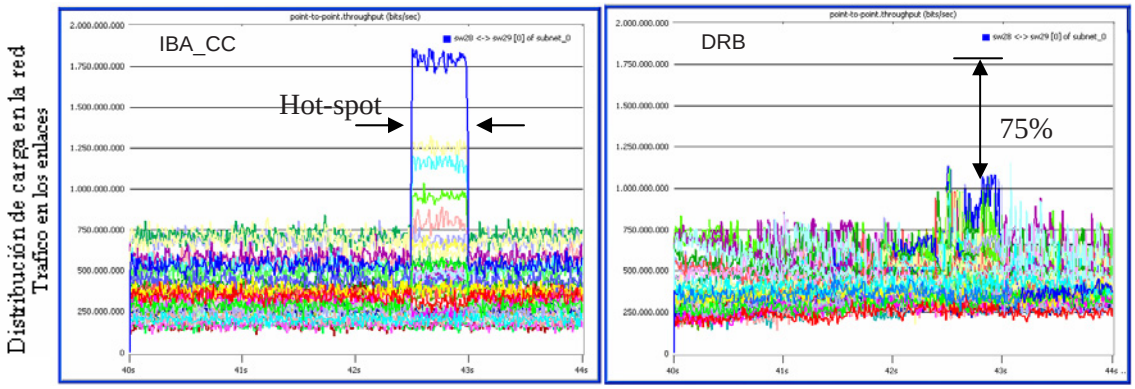


Figura 10. Distribución de carga en los enlaces de la red. Respuesta de los mecanismo al patrón hot-spot

En este análisis puede verse cómo la aplicación del algoritmo DRB mejora los resultados frente a la técnica de control de congestión utilizada por IBA en aproximadamente un *75%*. Cuando se utiliza DRB se eliminan efectivamente los picos de comunicación que se traducen en valores elevados de latencia, debido a que los paquetes deben esperar en los buffers a los que se conectan estos enlaces. DRB ofrece mejores resultados con patrones *"hot- spot"*, que presentan una gran concentración de carga local, ya que es capaz de distribuir el exceso de carga entre los diversos enlaces de la red y balancear de manera eficiente el volumen de comunicación de toda la red de interconexión.

6 CONCLUSIONES

En este artículo se ha propuesto, diseñado y evaluado un nuevo esquema de control de congestión para redes InfiniBand. El mecanismo propuesto elimina la degradación de las prestaciones de la red de interconexión, provocada por la elevada demanda de recursos en una determinada zona de la red. La eliminación del fenómeno de congestión se realiza distribuyendo el tráfico a través de trayectorias alternativas, lo que permite quitar carga en la zona congestionada. A diferencia de las técnicas que utilizan la regulación de mensajes (como la definida por IBA), el balanceo distribuido

del encaminamiento permite mantener la velocidad de inyección de mensajes lo que deriva en un aumento importante de la utilización de la red. Según se ha visto en la experimentación, este aumento está entre el *35* y el *50%* para el *throughput* y entre el *200* y el *300%* para la latencia. El mecanismo basa su funcionamiento en el uso conjunto de dos componentes independientes definidos en el modelo de gestión que especifica la arquitectura IBA. En primer lugar se configura el gestor de subred *(SM)* para que realice el descubrimiento de los componentes y asigne los identificadores locales y las mascaras que posibilitan el establecimiento de varias trayectorias alternativas entre el mismo par de nodos. Por otro lado se utilizan las capacidades de detección y notificación que ofrece el gestor de control de congestión *(CCM)*, y se desactiva la regulación de mensajes. De esta manera cuando se reciben mensajes de notificación, el mecanismo responde balanceando la carga en los enlaces de la red, mediante la selección de trayectorias alternativas. Debido a que ambos componentes de gestión están definidos en InfiniBand, nuestra propuesta es totalmente compatible y no requiere que se modifique ningún aspecto de la especificación. En el futuro tenemos planeado experimentar con diferentes patrones de tráfico y topologías más complejas, a fin de evaluar exhaustivamente a DRB y mejorar sus características intentando refinar la técnica de marcado de paquetes y el mecanismo de selección de trayectorias, teniendo en cuenta la sobrecarga provocada por los mensajes de notificación.

REFERENCIAS

[1] A Singh, WJ Dally, B Towles, AK Gupta 'Globally Adaptive Load-Balanced Routing on Tori', Computer Architecture Letters, IEEE, 2004.

[2] A. Vishnu, M. Koop, A. Moody, A. R. Mamidala, S. Narravula, D. K. Panda, "Hot-Spot Avoidance With Multi-Pathing Over InfiniBand: An MPI Perspective," ccgrid, pp. 479-486, 7º IEEE International Symposium on Cluster Computing and the Grid (CCGrid '07), 2007

[3] Baydal, E., 'A Family of Mechanisms for Congestion Control in Wormhole Networks', IEEE Trans. Parallel Distrib. Syst. 16(9), pp.772--784. 2005

[4] Duato, J. Johnson, I. Flich, J. Naven, F. Garcia, P. Nachiondo, T. "A new scalable and cost-effective congestion management strategy for lossless multistage interconnection networks" High-Performance Computer Architecture, HPCA-11. 2005

[5] Franco, D.; Garcés, I. & Luque, E., 'Avoiding Communication Hot-Spots in Interconnection Networks', in 'HICSS '99: Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences-Volume 8', IEEE Computer Society. 1999

[6] G. Pfister et al. "Solving Hot Spot Contention Using Infiniband Architecture Congestion Control". *Ion HPI-DC*, 2005.

[7] 'InfiniBand Architecture Specification' (v. 1.2), InfiniBand Trade Association. Disponible en: http://www.InfiniBandta.com/, 2004

[8] '*Opnet Modeler Accelerating Network R & D'* OPNET Technologies, Inc.,at http://opnet.com

[9] Santos, J.R.; Turner, Y.; Janakiraman G. 'End-to-end congestion control for infiniband' Infocom 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, 2003

[10] Shihang Yan, Geyong Min, Irfan Awan, "An Enhanced Congestion Control Mechanism in InfiniBand Networks for High Performance Computing Systems," aina, pp. 845-850, 20th International Conference on Advanced Information Networking and Applications. AINA 2006

[11] *'Top500 Supercomputers Site'*, at http://www.top500.org. Consultado el 19/07/2007.

[12] William Dally, Brian towles. Principles and practices of interconnection networks. Morgan Kaufmann publishers. 2004.

[13] Xuan-Yi Lin; Yeh-Ching Chung; Tai-Yi Huang**. '**A multiple LID routing scheme for fat-tree-based InfiniBand networks'**.** Parallel and Distributed Processing Symposium, Proceedings. 18th International, pp. 26-30, 2004.

Cómputo Paralelo Interclusters: Herramientas y Evaluación de Rendimiento

Fernando L. Romero, Antonio Quijano, Armando De Giusti[†], Fernando G. Tinetti[‡]

III-LIDI, Facultad de Informática, UNLP
50 y 115, 1900, La Plata, Argentina

CeTAD, Facultad de Ingeniería, UNLP
48 y 116, 1900, La Plata, Argentina

fromero@lidi.info.unlp.edu.ar, quijano@ing.unlp.edu.ar, degiusti@lidi.info.unlp.edu.ar, fernando@info.unlp.edu.ar

Resumen

Se presenta en este artículo la experiencia desarrollada referente a la utilización de múltiples clusters para cómputo paralelo. Inicialmente, se presentan algunas consideraciones importantes en cuanto la instalación y configuración del *middleware* o software de soporte necesario para realizar cómputo paralelo en este tipo de arquitectura que es relativamente *nueva*. Una de las premisas que se tiene en cuenta en este contexto es la de utilizar software herramientas relativamente estándar y estable. Una vez que se establece la configuración básica para ejecutar programas paralelos interclusters (en más de un cluster), se presentan los resultados obtenidos en lo referente a la caracterización de las transferencias de datos entre los clusters intervinientes. El rendimiento de las comunicaciones interclusters puede ser considerado como una de las características propias de este tipo de plataformas de cómputo paraelo y, por lo tanto, es importante tener una metodología y/o herramienta de caracterización de las mismas. Finalmente, se presentan algunos resultados interesantes de paralelización de dos aplicaciones sencillas a ejecutarse en computadoras de diferentes clusters. Esta paralelización muestra, por un lado, ejemplos específicos *reales* de ejecución programas paralelos intercluster y por otro lado, la aplicación de algunas ideas preexistentes de balance de carga que son aplicables también en este contexto.

Palabras claves: Comunicación de Procesos, Caracterización de Rendimiento, Sistemas Paralelos y Distribuidos, Paralelismo en Clusters e Interclusters.

Abstract

This paper presents the experience developed using multiple clusters for parallel computing. Initially, some relevant considerations are presented in terms installation and configuration of middleware or supporting software necessary to carry out parallel computing over this relatively new type of architecture. One of the premises taken into account within this context is that of using relatively standard and stable software and tools. Once the basic configuration is established to run interclusters (on more than one cluster) parallel programs, the obtained results regarding the characterization of data transferences among clusters are presented. Interclusters communication performance can be considered as one of the typical characteristics of this type of parallel computing platforms and, thus, it is important to count with a characterization methodology and/or tool of them. Finally, some interesting results of parallelizing two simple applications run in machines of different clusters are presented. On the one hand, this parallelization shows actual and specific results of running interclusters parallel programs and, on the other, the application of some pre-existing ideas of load balance, which are also applicable within this context.

Keywords: Process Communication, Performance Characterization, Parallel and Distributed Systems, Parallelism in Clusters and Interclusters.

[†] Investigador Principal CONICET. Profesor Titular D.E. Facultad de Informática UNLP
[‡] Investigador Asistente CICPBA

1 INTRODUCCION

La utilización de clusters para cómputo paralelo está bien establecida desde hace varios años [3] y actualmente existe una gran cantidad de clusters instalados que están siendo utilizados con software paralelo en producción. El modelo de programación de pasaje de mensajes también ha madurado en definiciones tales como las de PVM (Parallel Virtual Machine) [5] y MPI (Message Passing Interface) [14]. La biblioteca MPI se convirtió rápidamente en uno de los estándares más importantes para el desarrollo y la ejecución de programas paralelos. Esta biblioteca rápidamente se ha implementado en los sistemas de cómputo paralelo más distribuidos o desacoplados como lo son los clusters. De hecho, desde hace varios años existen al menos dos implementaciones de uso libre: MPICH [11] y LAM/MPI [4].

Aunque se puede asociar pasaje de mensajes y PVM y MPI en particular con hardware de cómputo distribuido o MIMD (Multiple Instruction Stream, Multiple Data Stream), PVM y MPI no *imponen* o *presuponen* ninguna característica sobre/de el hardware. Esto permite una gran independencia y portabilidad de los programas paralelos, incluyendo variantes como las de clusters de SMP (Symmetric MultiProcessing) y computadoras paralelas de memoria compartida [15]. En general, las bibliotecas de pasaje de mensajes resuelven básicamente dos problemas técnicos de manera satisfactoria para ser utilizadas:

1. Identificación única de procesos. Los programas paralelos no son mucho más que un conjunto de procesos que se pueden identificar de manera unívoca, independientemente de que se ejecuten en un ambiente de memoria compartida o memoria distribuida, por ejemplo.
2. Transferencia de datos entre los procesos. Si bien se reconoce que las primitivas básica de comunicación son del tipo *send*() y *receive*() punto a punto entre dos procesos, también se suelen incluir otras variantes como las comunicaciones colectivas del tipo de *broadcast*(), por ejemplo, o variantes semánticas de las comunicaciones punto a punto (haciendo alusión explícita a *buffers* de memoria, por ejemplo).

De hecho, en cualquier plataforma de cómputo donde se pueden resolver estos problemas, se podría utilizar una implementación de MPI, por ejemplo. Evidentemente los clusters han sido apropiados desde la perspectiva de las implementaciones de bibliotecas de pasaje de mensajes para cómputo paralelo, y estos dos problemas se resuelven de manera relativamente sencilla. Todo lo relacionado con transferencias de comunicaciones se ha resuelto utilizando protocolos estándares como TCP/IP y no es necesario un gran esfuerzo para asignar y mantener una identificación de procesos para bibliotecas como la más utilizada actualmente de MPI 1.1, donde no hay creación dinámica de procesos. De hecho, PVM también incluye la creación dinámica de procesos y tampoco en este caso es un gran problema mantener actualizada la asociación proceso-identificador en un ambiente distribuido. Evidentemente no es un problema en un ambiente centralizado (como la de un SMP), donde hay un único sistema operativo y, de hecho, cualquier sistema operativo tiene identificación única de procesos además de muchos otros datos de información de estado del sistema.

La propuesta en este artículo es la de plantear la solución al problema planteado por la ejecución de un programa paralelo en más de un cluster o, lo que es igual: la utilización de más de un cluster para ejecutar un programa paralelo, que se considerará sinónimo de “cómputo paralelo interclusters”. De una manera o de otra, hay varias alternativas de solución a este problema, algunas de ellas disponibles desde hace bastante tiempo. De hecho, se podría resolver inicialmente con las mismas bibliotecas PVM o implementaciones de MPI si no fuera por los controles de seguridad que se imponen actualmente al tráfico sobre Internet de la mayoría (si no todas) de las instituciones que tienen clusters disponibles para cómputo paralelo. De hecho, la propuesta avanza un poco más en el sentido de no solamente utilizar más de un cluster para un programa paralelo sino de hacerlo con el mínimo costo de instalación y mantenimiento de software y, además, con lo que se supone la menor

sobrecarga (*overhead*) de cómputo y comunicaciones posible. Justamente desde esta perspectiva se analizarán las propuestas existentes (al menos las más importantes o consideradas suficientemente representativas) en la siguiente sección.

Una vez resuelto el problema básico de cómputo paralelo interclusters (con la utilización de más de un cluster) comienza, en realidad, un problema mayor desde la perspectiva de paralelización de aplicaciones y la optimización de rendimiento. En un ambiente de cómputo paralelo interclusters no se puede asumir que todas las comunicaciones punto a punto tienen el mismo rendimiento, por ejemplo. Desde la perspectiva de un proceso que envía, algunos receptores son locales (en la misma red local, el mismo cluster) y otros no, están en otro cluster, al que se llega, en el caso más *usual*, por ruteo IP (normalmente). En cualquier caso, el rendimiento de las comunicaciones (el tiempo necesario para efectuar una transferencia de datos) no será el mismo. Otra de las características inherentes del cómputo paralelo interclusters es la de heterogeneidad de las computadoras a utilizar. Si bien en un cluster normalmente todas las computadoras son iguales (y, en particular, con la misma potencia de cálculo), es muy poco probable que las computadoras de dos o más clusters sean exactamente iguales entre sí. Y esto afecta directamente el balance de carga para que el cómputo sea *equilibrado* en cuanto al tiempo necesario en cada una de las computadoras que se utilizan. Los problemas planteados en cuanto a rendimiento de comunicaciones y balance de carga computacional (y tiempo de cómputo asociado) serán analizados en otra sección de este mismo artículo.

2 TRABAJOS PREVIOS RELACIONADOS

Aunque la terminología varía bastante, dado que no ha sido estandarizada, entre los primeros esfuerzos de utilizar computadoras en más de un cluster se puede encontrar la idea de “clusters geográficamente distribuidos”. Entre los primeros esfuerzos se pueden encontrar herramientas o bibliotecas tales como MagPIe del proyecto Albatros [24]. La biblioteca MagPIe fue orientada directamente a la optimización de las funciones de comunicaciones colectivas de MPI en redes de área extensa o extendida (WAN: Wide Area Network) [13] [12]. En algunos otros casos, se pueden encontrar los esfuerzos por hacer diferentes versiones de MPI interoperables, asumiendo que se tienen varios clusters, cada uno con su implementación *propia* (o propietaria o específicamente orientada a un tipo de máquina paralela o cluster). Entre estas iniciativas se pueden mencionar MPI-Connect [7] con su precedente PVMPI [6] y también IMPI [10], que se propuso con mayor generalidad para la interoperabilidad, con la definición de un protocolo apropiado.

La gran mayoría de las propuestas son de finales de la década de 1990, mayormente entre los años 1995 y 2000. Quizás por esta razón, de alguna manera o de otra estas propuestas han sido *absorbidas* o incluidas es lo que hoy se conoce como en Grid Computing [9]. Por ejemplo, el proyecto Albatross se asocia directamente con DAS [24] y el sitio web de DAS indica que ha sido sucedido por DAS-2 y DAS-3 [DAS-3], donde este último (DAS-3) se define como la infraestructura de Grid Computing en Holanda. Sin embargo, parece conveniente en ciertas aplicaciones mantener la propuesta de cómputo paralelo interclusters separado de Grid Computing por al menos dos razones:

- Grid Computing está propuesto como una solución *integral* o *completa* para compartir recursos de cómputo y almacenamiento a gran escala de distribución y de capacidad. Obviamente esto incluye la utilización de varios clusters, pero desde la perspectiva de compartir de manera controlada múltiples recursos disponibles en múltiples instituciones y/o instalaciones de cómputo. Esto incluye también la idea de mantener un gran sistema distribuido más que un gran sistema paralelo (aunque el sistema distribuido pueda utilizarse para cómputo paralelo).
- Grid Computing tiene mucho más que lo necesario para cómputo paralelo interclusters. Por

ejemplo, se intenta proveer SSI (Single System Image) no solamente a nivel de *Single Sign On* (o único punto de identificación y conexión) sino a nivel de proveer y obtener recursos para o de todo el sistema de Grid Computing. Esto implica, por ejemplo, un sistema muy elaborado y complejo para proveer adaptación a los diferentes sistemas de seguridad que se utilicen localmente en cada instalación/institución conectada a grid.

La propuesta de este artículo consiste en mantener la posibilidad de utilizar computadoras de más de un cluster sin tener que instalar toda una infraestructura (o *middleware*) como la de Globus [8]. Un ejemplo sencillo puede ser la colaboración puntual de dos o más instituciones para resolver un problema específico, donde esto no implique la definición e instalación de toda una infraestructura de software específica para el desarrollo de la solución, además de lo que realmente es necesario: desarrollar la solución. Cada una de las instituciones (o, más específicamente, cada uno de los clusters a utilizar) no debería ser mayormente afectado en cuanto a infraestructura de software para la colaboración en cuanto a cómputo paralelo. En cierta forma, puede considerarse que esto *restringe* o *limita* la escalabilidad de aplicaciones, pero sin lugar a dudas también reduce la complejidad de instalación y mantenimiento de la infraestructura de software necesaria. En el otro extremo, se podría mencionar la colaboración explícitamente *ad hoc* como en [1], en el sentido de desarrollar no solamente la aplicación sino también las comunicaciones y el control de la aplicación paralela. Aunque sin lugar a dudas esta es la forma con menor requerimiento de infraestructura de software *a priori*, también involucra un alto costo de desarrollo *extra* sobre el programador. Las comunicaciones, por ejemplo, se deberían llevar a cabo usando métodos que son relativamente rudimentarios para cómputo paralelo como el desarrollo directo sobre la biblioteca de *sockets BSD*. En este contexto, el objetivo es mantener un mínimo estable de desarrollo y ejecución de programas paralelos como el de MPI sin tener que recurrir a sus implementaciones para Grid Computing, por ejemplo, donde se requiere, además, todo el soporte que corresponde, como el que provee específicamente Globus.

Desde hace algún tiempo, se han estudiado características específicas de cómputo paralelo interclusters tales como la del problema generado por las interconexiones no dedicadas y los problemas de seguridad involucrados [2] [18]. A modo de resumen, en estos trabajos previos se reportan algunos detalles técnicos importantes a tener en cuenta para cómputo paralelo interclusters:

- En los ambientes no dedicados, muchos de los problemas de disponibilidad de las computadoras de los clusters a utilizar son propios de la falta de control sobre los mismos, no de las comunicaciones o estabilidad de las computadoras.
- Aunque la interconexión de los clusters no es dedicada, siempre hay conectividad entre los clusters a utilizar, salvo algunas excepciones relativamente muy poco frecuentes. Esto se debe básicamente a que la interconexión está involucrada con el tráfico de Internet, que es mantenido y monitoreado independientemente de la utilización de cómputo paralelo interclusters.
- El rendimiento que se podría considerar como *raw* (medido a partir de tráfico ICMP, por ejemplo) de las comunicaciones entre los clusters no es constante dado que no es dedicado, pero en general es muy cercano al máximo absoluto, al menos para transferencias de relativamente pocos datos (decenas de K Bytes, por ejemplo). Como es de esperar, el rendimiento para las comunicaciones interclusters fluctúa dependiendo de días y horarios.
- Los mecanismos básicos de seguridad que se imponen en las instituciones (y que son de uso común en casi todas las instalaciones de computadoras) normalmente impiden la utilización directa de implementaciones de MPI como MPICH y LAM/MPI. Estas implementaciones imponen un patrón de tráfico TCP/IP que normalmente es cancelado por los *firewalls* y/o mecanismos de seguridad de las instituciones.

3 UN SOPORTE SIMPLE PARA COMPUTO PARALELO INTERCLUSTERS

Quizás el soporte más simple para el cómputo paralelo interclusters es el que proveen las propias implementaciones de MPI que, *a priori*, permiten la utilización de múltiples computadoras independientemente de su ubicación geográfica y/o en clusters. Esta posibilidad debe ser descartada de plano por los múltiples problemas de seguridad y administración que involucra. Desde la perspectiva de seguridad, se deberían *relajar* los niveles de control, al menos desde la perspectiva de los *firewalls* que son de uso extendido en todas las instituciones. Como se reporta en [17], tanto el o los protocolos (TCP y/o UDP) como la cantidad y el *tipo* o *clase* (privilegiados o no) de *puertos* a utilizar por las aplicaciones que utilizan implementaciones de MPI no es configurable. Esto llevaría a dejar sin protección o sin control el tráfico entre las computadoras involucradas de los diferentes clusters. Aún en el caso de considerar que eliminar el control de tráfico entre las máquinas no es suficientemente peligroso, sí es un problema de administración. Al menos un administrador por institución debe encargarse de quitar estos controles para que las aplicaciones MPI puedan ser ejecutadas. Este problema se suele complicar debido a que normalmente los administradores de la seguridad no tienen relación directa con cómputo paralelo y que en las instituciones con control más distribuido se involucra a más de un administrador. Por lo tanto, mantener el uso directo de MPI no es sustentable desde el punto de vista técnico por la seguridad ni operativo por la necesidad de recurrir a múltiples administradores de las múltiples instituciones involucradas.

La *siguiente* alternativa para ejecutar aplicaciones paralelas interclusters basadas en MPI podría ser la utilización de algunas de las herramientas enumeradas en la sección anterior: MagPIE, MPI-Connect, PVMPI, o IMPI. Algunas de las alternativas podrían considerarse con el *costo* agregado de instalación de la o las bibliotecas más la recompilación de los programas paralelos. Sin embargo, como se comentó en la sección anterior, el problema más significativo es que la mayoría (sino todas) de estas herramientas simplemente se *trasladaron* al ambiente de Grid Computing. Esto significa que, como mínimo no tienen soporte ni actualizaciones para los clusters (o interclusters) actuales. De hecho, en el contexto de Grid Computing directamente existen implementaciones de MPI que evitarían el uso de otra biblioteca. Como también se comenta en la sección anterior, el objetivo es evitar la instalación de la infraestructura de una implementación para Grid Computing por el propio costo de la instalación y además para evitar el costo en tiempo de ejecución para el acceso a recursos (CPU, memoria, etc.) vía Grid Computing.

Tal como se adelanta en [18] a nivel preliminar, la idea es recurrir a la utilización de una red privada virtual, o VPN (*Virtual Private Network*) como *middleware* asociado en cierta forma a la implementación de MPI que se utilice. Desde un punto de vista técnico, todo lo que *necesita* MPI para ser utilizado en un cluster es que se tenga conectividad TCP/IP como en una red local. Esto es, justamente, lo que provee una VPN en cualquiera de sus versiones o implementaciones [25] [23]. Sin lugar a dudas, se tienen costos por la utilización de una implementación de VPN para cómputo paralelo interclusters:

- Se debe instalar y configurar el software de la implementación de VPN que se haya elegido. Normalmente no es muy complejo ni requiere muchos conocimientos previos. Casi todas las distribuciones de Linux tienen o incluyen alguna implementación. Este es el caso de OpenVPN, que es el que se utilizó para el trabajo de este artículo y cuya tarea de instalación y configuración se reporta en [19].
- Desde la perspectiva de seguridad o en realidad lo que se puede ver afectado por las controles de seguridad existentes vía *firewalls*, las implementaciones de VPN no impone mayores complicaciones. Normalmente se tiene un esquema cliente/servidor con un puerto bien conocido para el servidor, al cual los clientes hacen requerimientos. Esto implica que todo el

tráfico se "concentra" (de allí en parte el nombre de "túnel" o "entubamiento" o *tunnel* que suele aparecer en la bibliografía de VPN) en un puerto bien conocido del servidor que se instale. Esto de hecho simplifica lo relacionado con la seguridad, dado que la configuración de los *firewalls* se orienta, justamente, al control sobre números de IP y puertos a ser utilizados en tráfico TCP/UDP sobre IP. Al establecer un único IP (el del servidor) y un único puerto (el puerto bien conocido del servicio de VPN, que de hecho es configurable) las tareas son mínimas. De hecho, los controles actuales de tráfico *peer to peer* se pueden evitar con cierta sencillez dado que el servidor de VPN se puede configurar para que utilice puertos privilegiados en vez de los no privilegiados que normalmente se utilizan y se filtran en los *firewalls* por el tráfico *peer to peer*.

- Evidentemente en tiempo de ejecución existe una sobrecarga de procesamiento y en cierto modo también de tráfico de datos por el entubamiento de las comunicaciones sobre el esquema de cliente/servidor sobre un puerto bien conocido. En principio, en este artículo se pondrá énfasis en la factibilidad, por lo tanto no se estudiará específicamente el problema de la sobrecarga. Sin embargo, se puede estimar a priori que la utilización de una VPN implica menor sobrecarga que una infraestructura como la de Grid Computing.

La Fig. 1 muestra esquemáticamente el ambiente de ejecución de programas con MPI a partir de la utilización de una VPN. Desde la perspectiva del programador y sin tener en cuenta lo relacionado con el rendimiento de las comunicaciones, no hay ningún cambio, ni siquiera es necesario recompilar, dado que los binarios generados son independientes de que se utilice VPN o se ejecute en un cluster de computadoras en una LAN. En la Fig. 1 no se muestran las conexiones *reales* (las que resuelve la implementación de VPN instalada) entre las máquinas sino que con líneas de punto se muestran las conexiones vía VPN, como se utilizan desde los programas con MPI. El tráfico entre las computadoras de diferentes clusters se resuelve normalmente vía el que se define como *servidor* de VPN, dependiendo de la implementación y la configuración utilizadas.

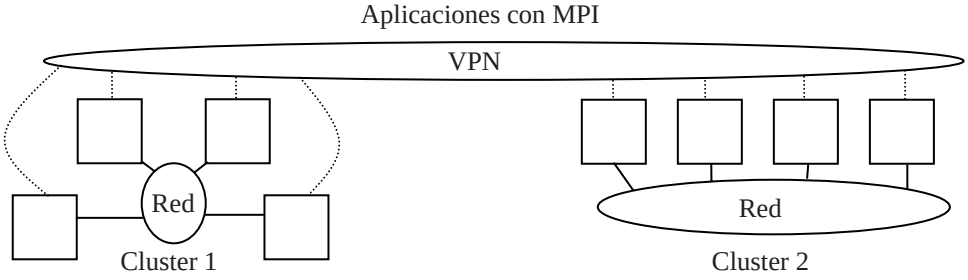


Figura 1: Infraestructura Sencilla para Cómputo Paralelo Interclusters con VPN.

4 EXPERIMENTACION: EVALUACION DE COMUNICACIONES

Tal como se reporta en [18], se utiliza el método sencillo de ping-pong de mensajes para evaluar el rendimiento de las comunicaciones entre dos procesos (en este caso en dos clusters diferentes), siguiendo el modelo de tiempo para las transferencias de datos

$$t(n) = \alpha + \beta\, n$$

donde n es la cantidad de datos, α es el tiempo de latencia (mínimo tiempo siempre se necesita para las comunicaciones) y β es la inversa del ancho de banda, o costo en tiempo por dato a comunicar. Uno de los primeros problemas encontrados, fue que no es posible utilizar cualquiera de las implementaciones de MPI sobre la VPN. En particular, LAM/MPI tiene problemas de ejecución, probablemente por las transferencias de datos entre los procesos lamd que ejecuta en cada una de las máquinas utilizadas. Para evitar el estudio de las razones de este problema, se recurrió a una implementación de MPI totalmente *estática* en el sentido de que no hay procesos de *administración*

o *intermedios* (del tipo de los lamd para LAM/MPI) para la ejecución de aplicaciones, MPICH, versión 1.2.4. Si bien en cierta forma se contradice la idea expresada en cuanto a que cualquier implementación de MPI se podría usar sobre la VPN, siempre es posible:

1. Encontrar el problema por el cual no funciona una implementación en particular y resolverlo (o hacer el requerimiento a quienes producen la implementación).
2. Encontrar una implementación de MPI que funcione sobre la VPN. En este caso, al menos para se recurre a MPICH que es de las implementaciones de MPI más utilizadas y respetadas de entre las de uso libre.

El entorno de experimentación elegido fue el más sencillo posible en cuanto a cantidades de computadoras: dos PCs con Linux, cada una en una red local diferente. Las dos redes locales involucradas son en realidad dos subredes de la 163.10.xx.yy de la UNLP, una en el ámbito de la Facultad de Informática y la otra en el ámbito de la Facultad de Ingeniería. La red de interconexión entre las dos redes locales no es exclusiva y, por lo tanto, se comparte o compite por el ancho de banda disponible entre estas redes con el tráfico usual de otras aplicaciones relacionado con Internet. La cantidad de computadoras con el que se comparte esta interconexión está en el orden de las centenas. La cantidad y el tráfico que involucran los experimentos ping-pong fueron determinados de forma tal que:

- No utilicen más del 5% de 10Mb/s que se asume como el máximo ancho de banda disponible entre las redes locales, con múltiples *routers* intermedios, algunos de los cuales utilizando placas Ethernet de 10 Mb/s. En cualquier caso, este tráfico es muy *conservador* en el sentido de evitar al máximo cualquier congestión de tráfico en la red de interconexión compartida.
- Los experimentos se distribuyen de manera uniforme durante todo el tiempo de ejecución, de forma tal que se monitoricen tiempos de uso normal de las redes locales y de tráfico de Internet con el que se compite con otras aplicaciones en el tramo de interconexión entre los clusters.
- El 50% de los experimentos se *orienta* a identificar la latencia de los mensajes entre dos procesos de una aplicación paralela y el otro 50% se *orienta* a la identificación del ancho de banda disponible o *posible* entre los procesos. La longitud de los mensajes de los experimentos orientados a latencia se establece en 8 bytes y la longitud de los mensajes orientados a ancho de banda se establece en 20000 bytes (básicamente para no provocar ráfagas de uso muy intensivo sobre la red compartida, como se explica en el primer punto).
- Los experimentos se llevaron a cabo durante aproximadamente 10 días corridos, para tener datos de días y horas de uso normal de las computadoras y las redes intermedias y también de días y horas con relativamente poco de uso de las computadoras y redes involucradas.

La Fig. 2 muestra el histograma de la distribución de los tiempos de latencia de los experimentos, donde se puede observar que la gran mayoría de los mensajes tiene una latencia de entre 1 y 5 milisegundos. Es claro que esta latencia es muy elevada para las capacidades de cómputo de las PCs actuales (que son del orden de Gflop/s), pero en cierto modo es muy importante identificar con experimentos tan sencillos que alrededor del 95% de las comunicaciones tienen valores de latencia en este rango. Esta información es particularmente útil para las aplicaciones paralelas, dado que dan una idea de la granularidad mínima: no tiene sentido comunicarse entre tiempos de cómputo menores a los de la latencia. La Fig. 3 muestra la distribución del ancho de banda de los experimentos realizados con 20000 bytes, donde como en el caso de la latencia se puede observar que la mayoría está concentrado en un rango de valores relativamente pequeño. A modo de resumen de los valores mostrados en la Fig. 3, la gran mayoría (alrededor de 95%) de los experimentos se lleva a cabo con entre 50 y 58 KB/s. Es de destacar que durante todo el tiempo de ejecución de los experimentos no hubo que reinstalar ni reconfigurar OpenVPN, la conectividad entre las computadoras se mantuvo dentro de la VPN, a diferencia de lo que había sucedido en los experimentos reportados en [20].

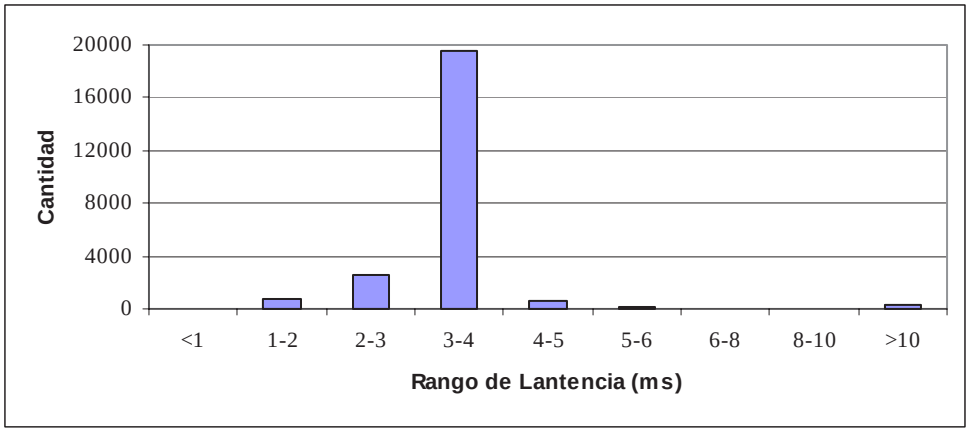


Figura 2: Distribución de Tiempos de Latencia (*Startup Time*) con MPI en VPN.

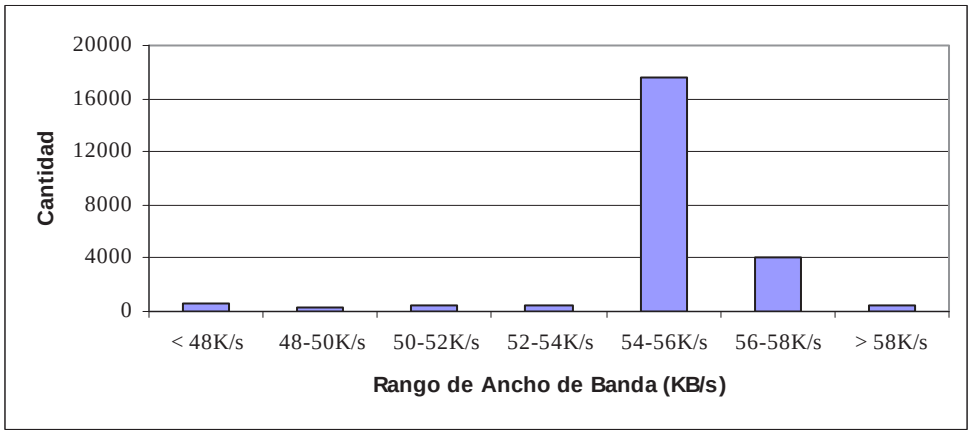


Figura 3: Distribución de Anchos de Banda con MPI en VPN.

5 EXPERIMENTACION: PARALELIZACION DE APLICACIONES

Como ya se ha comentado, el hecho de que sea factible el cómputo paralelo interclusters no resuelve los problemas asociados a la paralelización de aplicaciones. De hecho se podría afirmar que por un lado aporta un horizonte nuevo de paralelización y sus problemas asociados. Uno de ellos es la asimetría o diferencia en las comunicaciones *intra* e *inter* clusters que se ha comentado. A partir de los resultados que se muestran en la sección anterior se tienen datos más precisos al menos del entorno sobre el que se llevó a cabo la experimentación:

- El rendimiento de las comunicaciones interclusters no es constante, aunque los rangos son reducidos.
- El rendimiento de las comunicaciones interclusters es varias veces menor que el de las comunicaciones dentro de un mismo cluster. Solamente a modo de ejemplo, en un cluster interconectado con Ethernet de 100 Mb/s se tienen aproximadamente 10 MB/s entre procesos, lo cual es muy superior al rango de 50-58 KB/s que se muestra en la Fig. 3.

Las aplicaciones denominadas *altamente paralelas* o *embarazosamente paralelas* (*embarrassingly parallel*) son las primeras opciones para ser resueltas en cualquier plataforma de cómputo paralelo. Esto se debe a que en realidad paralelizar estas aplicaciones no requiere casi ningún esfuerzo más allá de la propia codificación del programa, por ejemplo con MPI. Por otro lado, estas aplicaciones normalmente requieren muy pocas comunicaciones durante el tiempo de cómputo o, puesto de otra forma, se pueden paralelizar con granularidad muy gruesa. Esto las hace *mejores* en cuanto a la obtención de rendimiento óptimo o cercano al óptimo en ambientes con muy bajo rendimiento de

comunicaciones, como el que se utiliza para la experimentación en este artículo.

Se han elegido dos aplicaciones muy conocidas pero con diferentes características de paralelización con la finalidad de evaluar rendimiento: integración numérica (cálculo del número π) y cálculos asociados con el conjunto de Mandelbrot. En ambos casos, los problemas de cálculo asociados son muy sencillos de paralelizar dado que se pueden identificar *partes* de cómputo independiente de manera natural, sin recurrir a cambios en el tipo de cómputo secuencial básico. Nuevamente el ambiente de experimentación es el de mínima complejidad en cuanto a cantidad de computadoras: una computadora en cada uno de los clusters. Dado que es importante la capacidad de cómputo para estos experimentos, en la Tabla 1 se detallan las características técnicas más importantes de las dos computadoras utilizadas, donde se puede notar que son muy diferentes en cuanto a capacidad de procesamiento y almacenamiento.

Tipo	CPU	Memoria RAM	Sistema Operativo
PC	Pentium 4 2.4 GHz	1 GB	Linux 2.4.18-14
PC	Pentium II 400 MHz	128 MB	Linux 2.4.18-14

Tabla 1: Características de las Computadoras Utilizadas en los Experimentos.

Un problema común en el contexto de cómputo paralelo interclusters es el de la heterogeneidad en cuanto a la capacidad de cómputo de las computadoras que se utilizan. Desde la perspectiva de la paralelización, esto genera un problema de balance de carga implícito en la suposición de que a todos los procesadores se les asignará la misma cantidad de operaciones a realizar. Se debe notar que esto no es un problema de paralelización sino de rendimiento (quizás implícito) de la paralelización. Para balancear la carga o, más específicamente, para asignar la carga de procesamiento acorde a la capacidad relativa de cada procesador se tienen que resolver dos problemas:

1. Conocer las diferencias relativas de capacidades de cómputo, es decir para cada par de computadoras cuánto mejor o peor es una respecto de la otra.
2. Asignar la cantidad de cómputo de cada computadora de acuerdo con su capacidad relativa respecto de las demás.

El primer problema es sencillo de resolver utilizando la idea ya planteada en [21] y [16], es decir ejecutando el mismo problema pero con tamaño reducido en cada una de las computadoras y relacionando directamente los tiempos de cómputo. Con estos cálculos no solamente se puede definir el balance de carga sino también la evaluación de rendimiento. El segundo de los problemas planteado normalmente depende de la aplicación, es decir que se resuelve caso por caso.

5.1 Cálculo de π

Una forma numérica de calcular el número **π** se lleva a cabo vía integración numérica con el cálculo

$$\pi = h \times \sum_{i=0}^{n-1} 4/(1 + x_i^2) \qquad (1)$$

donde $h = 1/n$, n es la cantidad de puntos a utilizar en la integración numérica y $x_i = h \times (i + 0.5)$. En general, a mayor cantidad de puntos (n mayor) es *mejor* o *más precisa* la aproximación de π que se obtiene. Claramente, cada término de la sumatoria puede ser calculado de manera absolutamente independiente de todos los demás y, por lo tanto, estos cálculos (y las sumas intermedias de los mismos) pueden ser paralelizados en tantos procesadores como se decida. Si bien la paralelización es trivial, no sucede lo mismo con el balance de carga, dado que no se logra balancear la carga en *p*

procesadores realizando la suma parcial de *n*/*p* términos de la Ec. (1) en cada uno de ellos. En este caso particular, se puede aprovechar de manera directa otra de las características del cálculo planteado en la Ec. (1): todos los términos de la sumatoria implican el mismo costo en términos de operaciones numéricas. En este sentido, el problema no solamente es sencillo en cuanto a su división en partes sino que también es sencillo en cuanto a balance de carga en ambientes heterogéneos: la cantidad de términos de la sumatoria de la Ec. (1) a resolver en cada computadora es directamente proporcional a su potencia de cómputo relativa. En el caso de las PCs de la Tabla 1, la capacidad de cómputo relativa es tal que una es casi tres veces *mejor* que la otra en términos de capacidad de cálculo para π. Los resultados de los experimentos realizados con el cálculo de π se resumen en la Tabla 2, donde se muestran las velocidades relativas normalizadas con respecto a la de mayor capacidad de cómputo.

Velocidades Normalizadas	1 y 0,36
Eficiencia de la Paralelización	94%

Tabla 2: Resumen de la Experimentación con el Cálculo de π Interclusters.

Aunque la cantidad de computadoras es la mínima, el hecho de obtener una eficiencia del 94% es muy satisfactorio dado el bajo rendimiento de las interconexiones entre los clusters utilizados. En el caso de estas aplicaciones muy sencillas de paralelizar y balancear en cuanto a carga de trabajo, es de esperar que la eficiencia se mantenga alta aumentando la cantidad de computadoras.

5.2 Cálculos Asociados al Conjunto de Mandelbrot

La Fig. 4 muestra una de las formas más comunes de cómputo utilizado para el gráfico relacionado con el conjunto de Mandelbrot (mencionado en algunos casos como "*escape time algorithm*" [26]) para un punto dado como (x0, y0) [22].

```
x = x0; y = y0; iter = 0;
while ( x*x + y*y < (2*2)  AND  iter < max)
{
   xtemp = x*x - y*y + x0; y = 2*x*y + y0;
   x = xtemp; iter = iter + 1;
}
color = iter
```

Figura 4: Cómputo Relacionado con el Conjunto de Mandelbrot.

Comparándolo con el cómputo de π anterior, el que se muestra en la Fig. 4:

- Es similar en cuanto a que el valor de un punto en particular es totalmente independiente de todos los demás valores.
- Es diferente en cuanto a que la cantidad de operaciones necesarias para el cálculo del valor de un punto en particular no es conocida a priori, depende del punto mismo (iteración *while*).

Normalmente, lo que se paraleliza es el cálculo de los diferentes puntos y evidentemente no requiere mucho esfuerzo. No es tan directo el balance de carga, aún cuando se conozcan las velocidades relativas de las computadoras a utilizar. Tampoco es demasiado complejo, dado que se puede recurrir a un esquema similar al utilizado en cómputos de álgebra lineal, por ejemplo, donde se distribuye el cálculo de forma tal que cada proceso debe obtener valores relativamente dispersos en el espacio total a calcular [16]. En el caso de álgebra lineal, esto se aplica sobre la matriz a factorizar en L y U, por ejemplo, y en este caso se aplicará al conjunto de puntos a calcular. Visto como una matriz, el conjunto de puntos a calcular se puede dividir en múltiples bloques de relativamente pocas filas o columnas y estos bloques se asignan a las diferentes computadoras.

Teniendo una cantidad de bloques suficientemente grandes, la cantidad de bloques asignados a cada computadora es directamente proporcional a su velocidad relativa. En el caso específico del conjunto de Mandelbrot, se llevó a cabo el cálculo para un *espacio* de 800x800 puntos que se dividió en bloques de 100x800 puntos (bloques de 100 filas) y de cada uno de estos bloques, la cantidad de filas asignadas a cada computadora es proporcional a la velocidad relativa de las mismas. La Tabla 3 muestra el resumen de los experimentos realizados con este problema, aplicando el balance de carga que se describió previamente. Una vez más, la eficiencia de la paralelización es muy satisfactoria para la plataforma de cómputo subyacente.

Velocidades Normalizadas	1 y 0,36
Eficiencia de la Paralelización	89%

Tabla 2: Resumen de la Experimentación con el Cálculo de π Interclusters.

6 CONCLUSIONES Y TRABAJO FUTURO

En este artículo se ha mostrado la utilización de una VPN para cómputo paralelo interclusters utilizando una implementación específica de MPI. Esto muestra que, al menos en principio, no es necesaria una infraestructura muy compleja sino *middleware* estándar para cómputo paralelo interclusters. Los experimentos han mostrado que es posible caracterizar satisfactoriamente el rendimiento de las comunicaciones interclusters de manera sencilla, y a partir de esta información se pueden tomar decisiones importantes de paralelización. También se ha mostrado que no solamente es factible paralelizar aplicaciones, (al menos las del tipo *embarrassingly parallel*), sino que se puede obtener rendimiento muy satisfactorio, con eficiencia de la paralelización de alrededor del 90% o mayor. También se ha mostrado que las estrategias conocidas de balance de carga dan resultados satisfactorios, al menos en las aplicaciones que se mostraron.

Una de las extensiones inmediatas de este trabajo consiste en la utilización de más de una computadora en cada cluster involucrado. Sin lugar a dudas el objetivo final de cómputo paralelo interclusters debería ser el de utilizar todos los clusters de manera completa, para aprovechar al máximo la capacidad de cálculo disponible. En caso contrario, se debería establecer una estrategia o política de utilización de las computadoras de cada cluster que justifique el hecho de no utilizar las computadoras disponibles de los clusters.

REFERENCIAS

[1] E. Argollo, D. Rexachs, F. G. Tinetti, E. Luque, “Efficient Execution of Scientific Computation on Geographically Distributed Clusters”, Applied Parallel Computing: 7th International Conference, PARA 2004, Lyngby, Denmark, June 20-23, 2004, Revised Selected Papers, LNCS 3732. Editors: J. Dongarra, K. Madsen, J. Wasniewski, ISBN: 3-540-29067-2, Springer-Verlag Berlin Heidelberg 2006, pp. 691–698.
[2] W. Aróztegui, F. L. Romero, F. G. Tinetti, “Comunicaciones para Cómputo Paralelo Intercluster”, Anales del VIII Workshop de Investigadores en Ciencias de la Computación, Universidad de Morón, Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales, Morón, Argentina, Junio 1-2 de 2006, ISBN 950-9474-34-7, pp. 211-215.
[3] M. Baker, R. Buyya, “Cluster Computing at a Glance'”, en R. Buyya Ed., High Performance Cluster Computing: Architectures and Systems, Vol. 1, Prentice-Hall, Upper Saddle River, NJ, USA, pp. 3-47, 1999.
[4] G. Burns, R. Daoud, J. Vaigl, “LAM: An Open Cluster Environment for MPI”, Proc. of Supercomputing Symposium, pp. 379-386, 1994. Available at http://www.lammpi.org/download/files/lam-papers.tar.gz
[5] J. Dongarra, A. Geist, R. Manchek, V. Sunderam, “Integrated pvm framework supports heterogeneous network computing”, Computers in Physics, (7) 2, pp. 166-175, April 1993.
[6] G. Faag, J. Dongarra, A. Geist, “PVMPI provides interoperability between MPI implementations”, in

Proc. 8th SIAM Conf. on Parallel Processing, SIAM (1997).
[7] G. E. Fagg, K. S. London, "MPI interconnection and control", Technical Report Tech Rep. 98-42, Corps of Engineers Waterways Experiment Station Major Shared Resource Center (1998).
[8] I. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems", IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2006.
[9] I. Foster, C. Kesselman, The Grid 2: Blueprint for a New Computing Infrastructure, 2nd Edition, Morgan Kaufmann, 2003, ISBN 1558609334.
[10] W. L. George, J. G. Hagedorn, J. E. Devaney, "IMPI: Making MPI Interoperable", J. of Research of the National Institute of Standards and Technology, Volume 105, Number 3, May–June 2000, pp. 343-348.
[11] W. Gropp, E. Lusk, "Sowing MPICH: A Case Study in the Dissemination of a Portable Environment for Parallel Scientific Computing", The International Journal of Supercomputer Applications and High Performance Computing, Vol. 11, No. 2, pp. 103-114, Summer 1997,
[12] T. Kielmann, H. E. Bal, S. Gorlatch, K. Verstoep, R. F.H. Hofman, "Network performance-aware collective communication for clustered wide-area systems", Parallel Computing archive, Volume 27, Issue 11, Oct. 2001, Clusters and computational grids for scientific computing, pp. 1431 – 1456, ISSN 0167-8191.
[13] T. Kielmann, R. F. H. Hofman, H. E. Bal, A. Plaat, R. A. F. Bhoedjang, "MAGPIE: MPI's collective communication operations for clustered wide area systems", Seventh ACMSIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'99), Atlanta, GA, May 1999, pp. 131–140.
[14] MPI Forum, "MPI: a message-passing interface standard", International Journal of Supercomputer Applications, 8 (3/4), pp. 165-416, 1994.
[15] A. Tanenbaum, "Structured Computer Organization", 5th Ed., Prentice Hall, ISBN 0131485210, 2005.
[16] Fernando G. Tinetti, Tesis Doctoral, Cómputo Paralelo en Redes Locales de Computadoras, Tesis Doctoral, Univ. Autónoma de Barcelona, 2004. Disponible en http://ftinetti.googlepages.com/tesisdoctoral
[17] Tinetti F. G., Aróztegui W., "Bibliotecas de Pasaje de Mensajes y Cómputo Intercluster", Rep. Técnico PLA-003-2005, Sep. 2005. Disponible http://ftinetti.googlepages.com/ reportest%C3%A9cnicos2006
[18] F. G. Tinetti, W. J. Aróztegui, "Factibilidad y Rendimiento de las Comunicaciones para Cómputo Paralelo Intercluster", XII Congreso Argentino de Cencias de la Computación (CACIC), Oct. 17-21, 2006, Universidad Nacional de San Luis, Potrero de los Funes, San Luis, Argentina.
[19] F. G. Tinetti, W. Aróztegui, "Instalación y Configuración de OpenVPN 2.0 para Cómputo Paralelo Intercluster", Reporte Técnico PLA-002-2006, Julio 2006. Disponible en http://ftinetti.googlepages.com/ reportest%C3%A9cnicos2006
[20] F. G. Tinetti, W. Aróztegui, "Perfil Preliminar de las Comunicaciones Intercluster", Reporte Técnico PLA-001-2006, Marzo 2006. Disponible en http://ftinetti.googlepages.com/reportest%C3%A9cnicos2006
[21] F. G. Tinetti, A. Quijano, A. De Giusti, E. Luque, "Heterogeneous Networks of Workstations and the Parallel Matrix Multiplication", Recent Advances in Parallel Virtual Machine and Message Passing Interface, 8th European PVM/MPI, Santorini/Thera, Greece, Sep. 23-26, 2001, Proceedings, Y. Cotronis, J. Dongarra (Eds.). LNCS 2131 Springer 2001, ISBN 3-540-42609-4, pp. 296-303.
[22] B. Wilkinson, M. Allen, Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, 2nd Ed., Prentice Hall, 2004, ISBN 0131405632.
[23] J. Yonan, "Understanding the User-Space VPN: History, Conceptual Foundations, and Practical Usage", Linux Fest Northwest 2004. Disponible en http://openvpn.net/papers/BLUG-talk/BLUG-talk.ppt
[24] Albatross: Wide Area Cluster Computing Homepage, http://www.cs.vu.nl/albatross/
[DAS-3] The Distributed ASCI Supercomputer 3 (DAS-3) Homepage, http://www.cs.vu.nl/das3/
[25] Wikipedia, the free encyclopedia, "Virtual Private Networks", http://en.wikipedia.org/wiki/Virtual_ private_network
[26] Wikipedia, the free encyclopedia, "Mandelbrot set", http://en.wikipedia.org/wiki/Mandelbrot_set

Conceptos Fundamentales de Diseño en Sistemas de Programación Esqueletal

Fernando Saez Marcela Printista
LIDIC.
Universidad Nacional de San Luis.
Ejército de los Andes 950, San Luis, Argentina.
e-mail: {**bfsaez@unsl.edu.ar, mprinti@unsl.edu.ar**}

Abstract

In the last time the parallel programming community has worked to look for new templates or skeletons. A significant number of projects has built real systems, but none of these has achieved significant popularity neither developtment projects nor academic community. We reviewed some fundamentals concepts of design that should be supported by skeletal parallel programming systems to fortify their use and we present the development of a high level prototype adapts some of these essential concepts. This prototype resolves a high spectre of divide and conquer problems and their use is showed through three simple examples.

Keywords: Skeletal Programming System, Divide and Conquer, Design Concepts

Resumen

En los últimos tiempos, la comunidad de programación paralela ha trabajado mucho en soluciones basadas en patrones y esqueletos. Un número importante de proyectos han construido sistemas reales, pero ninguno de ellos ha alcanzado una popularidad notable en los entornos de desarrollo como tampoco en la comunidad científica paralela. El objetivo de este trabajo es revisar algunos conceptos fundamentales que deben ser tenidos en cuenta durante la etapa de diseño de patrones y que fortalecen el uso de un sistema de programación paralela esqueletal. Además, en este trabajo se presenta el desarrollo de un prototipo de alto nivel, el cual incorpora alguno de estos conceptos esenciales. Este prototipo resuelve un amplio espectro de problemas Divide y Vencerás y su uso es mostrado a través de tres ejemplos muy sencillos.

Palabras claves: Sistema de Programación Esqueletal, Divide y Vencerás, Conceptos de Diseño

1 INTRODUCCIÓN

Muchos algoritmos paralelos pueden ser caracterizados y clasificados por su adhesión a uno o más patrones de computación. La programación esqueletal propone que tales patrones sean abstraídos y provistos al programador como un conjunto de herramientas formando así un sistema de programación esqueletal. Este sistema de programación permite resolver un problema concreto a partir de la creación de instancias de métodos más generales. El sistema esta compuesto por un conjunto de

esqueletos (paradigmas) desde los cuales pueden ser derivados programas modelos que ilustren como el paradigma resuelve problemas específicos [5].

El objetivo del sistema esqueletal es reducir las dificultades del desarrollo de software por medio de la abstracción y la reusabilidad de programas paralelos y mejorar la performance dando acceso a los esqueletos optimizados en arquitecturas particulares.

A partir de la experiencia obtenida en la programación esqueletal, especialmente en el diseño de distintos esqueletos Divide y Vencerás desarrollados con propósitos específicos [7, 8, 9] y en base a lo presentado en [2], se ha podido interpretar conceptos fundamentales de diseño que deben ser soportados por un sistema de programación esqueletal. La aplicación de estos conceptos ha conducido al desarrollo de un primitivo de alto nivel cuyo objetivo es capturar estas características para que resulte altamente abstracto y funcional.

El objetivo de este trabajo es revisar y poner a discusión estos conceptos de diseño, mostrar el primitivo $D\&V$ desarrollado en forma de librería de funciones C soportadas en MPI [1] y poner en contexto la implementación del esqueleto $D\&V$ a través de ejemplos muy sencillos.

En la sección 2 se introduce una terminología clara y sencilla que permite definir cada uno de los conceptos, en la sección 3 se describe el esqueleto $D\&V$ implementado y se muestra como éste puede ser usado para resolver distintos tipos de problemas $D\&V$. Por último en la sección 4 se presentan las conclusiones y los trabajos futuros.

2 CONCEPTOS Y TERMINOLOGÍA

Con el propósito de formalizar las propiedades fundamentales de diseño que debe satisfacer un sistema de programación esqueletal, este trabajo adopta la terminología expuesta por Cole [2], la cual es clara y simplifica la comprensión de los conceptos.

El propósito de cualquier esqueleto es abstraer un patrón de *actividades* y de sus *interacciones*.

Entre los aspectos a ser evaluados cuando se identifican las actividades podemos identificar: el tamaño del gránulo, el grado de reusabilidad del código que implementa la actividad y el mapeo de las actividades a la arquitectura, entre otros.

En cuanto a las interacciones, es importante distinguir las interacciones, denominadas *internas*, que ocurren entre dos o más actividades. Estas son las encargadas de capturar el paradigma de computación paralela y la topología del grupo de procesos intervinientes. En una interacción interna, las actividades pueden estar asignadas a un mismo procesador o residir en distintos procesadores.

Por otro lado, las interacciones *externas* ocurren entre las actividades y el contexto que las contiene. Estas últimas serán las encargadas de la entrada-salida del primitivo. Por ejemplo, en una aplicación de procesamiento de imágenes debería ser posible administrar, de manera explícita, que distintos segmentos de una misma imagen que se encuentran almacenados en varios procesos conforman una única imagen.

Existe una clasificación que involucra el modo en cómo interactúa un esqueleto con su contexto y que está determinado por el origen y destino de la entrada y salida del esqueleto:

Centralizado-Centralizado (CC): es aquél cuyo conjunto de datos de entrada y de resultados se encuentran en un único proceso de todos los disponibles para la ejecución del esqueleto.

Distribuido-Distribuido (DD): es aquél cuyo conjunto de datos de entrada y de salida se encuentran distribuidos entre los distintos procesos.

Replicado-Replicado (RR): es aquél cuyo conjunto total de datos de entrada y de salida se encuentran replicados en cada uno de los procesos.

Las restantes combinaciones (CR, CD, DC, DR, RC, RD) pueden ser fácilmente verificadas.

Un problema de procesamiento de imágenes resuelto con el paradigma de programación pipeline permite ejemplificar la categoría CD. Suponga un problema cuyo objetivo es remover frecuencias específicas ($f_0, f_1, ..., f_{p-1}$) de una imagen digitalizada. Durante el procesamiento de imágenes es importante recuperar las imágenes intermedias que van surgiendo luego de la eliminación de cada f_i. Este problema puede ser resuelto por un pipeline CD de p etapas donde cada etapa del pipeline recibe una imagen ($DigI_i$), le aplica un filtro fil_i, obtiene una nueva imagen $DigI_{i+1}$ a la cual se le han eliminado las frecuencias $f_0, ..., f_i$, la almacena en un buffer de salida del proceso y la transmite a la siguiente etapa.

Otro ejemplo se puede observar en un pipeline que resuelve problemas denominados $All-Pairs$. Una computación $All-Pairs$ sobre un sistema de n elementos puede ser paralelizado por medio de un pipeline DD. Cada etapa contribuye con un subsistema de elementos y en ella se realiza una computación $All-Pairs$ sobre los elementos retenidos. Luego cada etapa recibe elementos de su vecino izquierdo, los hace interactuar con los elementos retenidos, los almacena en un buffer y los envía a su vecino derecho. Todos los elementos quedan igualmente distribuidos entre las etapas del pipeline. Una aplicación de este tipo de pipeline puede verse en [8].

Como se mencionó antes, un esqueleto debe proveer una abstracción de las interacciones entre las distintas actividades que conforman un programa paralelo, pero trae aparejado un proceso de estructuración que restringe el modo en que las actividades pueden interactuar. Existen dos tipos de restricciones:

- *Espaciales*: determinan que actividades pueden interactuar entre si (actividades socias) y especifican el rol (fuente y/o destino) de cada una en la interacción.
- *Temporales*: determinan el orden temporal correcto de interacción entre actividades socias.

Por ejemplo, un pipeline clásico es definido por una restricción espacial especificando que las interacciones se producirán entre actividades asignadas a dos etapas sucesivas. En otras palabras:

- Las actividades A_i y A_{i+1} son socias, para $0 \leq i < p-1$ y la interacción se producirá desde A_i hacia A_{i+1} (interacción: $I_{i,i+1}$).
- Las actividades A_i y A_{i-1} son socias, para $0 < i < p$ y la interacción se producirá desde A_{i-1} hacia A_i (interacción: $I_{i-1,i}$).

Una restricción temporal para este mismo ejemplo, debe determinar que durante el mismo ciclo de ejecución, toda etapa se comunique, en primer lugar, con la etapa anterior del pipeline y luego con la etapa posterior. En otras palabras, durante un mismo ciclo de ejecución, una actividad A_i deberá ejecutar la interacción $I_{i-1,i}$ y luego la interacción $I_{i,i+1}$, en ese orden.

En la próxima sección se describen algunos de los conceptos fundamentales que debe soportar un sistema de programación esqueletal para alcanzar un mayor grado de flexibilidad y permitir resolver una mayor diversidad de problemas.

2.1 Anidamiento

Informalmente el anidamiento de esqueletos es un término general usado para denotar una situación en donde dos o más esqueletos paralelos están activos en el mismo tiempo. El anidamiento de esqueletos es una propiedad importante que debe satisfacer un sistema esqueletal para poder combinar e integrar las distintas facilidades formando así estructuras no convencionales que permitan resolver algoritmos más complejos. También permite que el sistema simplifique la expresión de algoritmos

paralelos que operan sobre estructuras de datos irregulares, ya que permite expresiones directas de conceptos como “En paralelo, por cada vértice en un grafo, buscar su vecino mínimo” o “En paralelo, por cada fila en una matriz, sumar sus elementos”. En ambos casos las acciones internas (Buscar el vecino mínimo o sumar la fila) también pueden ser realizadas en paralelo.

Es posible identificar dos modos de anidamientos:

- *Transitorio*: una actividad podría invocar a otro esqueleto para procesar algún dato local o para realizar otra computación autónoma. Aquí se mantienen las interacciones externas entre el esqueleto anidado y la actividad invocante.
- *Persistente*: desde la perspectiva del esqueleto anidado, sus interacciones externas se convierten en interacciones internas entre la actividad invocante del esqueleto de nivel superior y el esqueleto anidado.

En el ejemplo del pipeline, una etapa formada por varios procesos podría recibir un dato, particionarlo en una colección de subdatos, procesarlos independientemente en otro pipeline interno y finalmente construir el resultado a entregar al nivel superior. Cada invocación del pipeline anidado es iniciada por una actividad del nivel superior.

En una actividad, bajo un anidamiento transitorio, la interacción externa se produce con una invocación a una función convencional (llamada al esqueleto anidado). En esta invocación, los datos de entrada deben ser explícitamente definidos por el programador. Al finalizar la ejecución del pipeline anidado, los datos de salida deben ser manipulados para construir la solución que se devolverá a la actividad invocante.

Por otro lado, la misma etapa del pipeline anterior podría invocar un pipeline persistente como su nivel interno, aquí cada dato requerido por la primera etapa del pipeline anidado interactúa internamente con la etapa anterior del pipeline de nivel superior. La invocación persistente del pipeline indica que los datos navegarán (interactuarán internamente) desde el esqueleto padre hacia el esqueleto hijo y luego nuevamente del hijo al esqueleto padre, por lo tanto, en este caso, no es necesario definir su entrada y salida.

2.2 Modo de Interacción

La estructura de un esqueleto especifica las actividades que lo conforman y las interacciones permitidas entre estas actividades. Siguiendo la terminología utilizada en [2], toda actividad realiza interacciones *implícitas* las cuales estan restringidas temporalmente por el esqueleto. Pero la estructura de un esqueleto también debería ser flexible y permitir que se produzcan interacciones no contempladas por la estructura y de esa manera atender necesidades excepcionales de la aplicación. El modo de interacción es el mecanismo que le permitirá al programador indicar la ocurrencia de interacciones *explícitas*. En este modo, las actividades realizan interacciones definidas específicamente en el código de una actividad. Este último modo debe ser manejado por el sistema de una forma cuidadosa, ya que si bien es ofrecido como una facilidad del esqueleto, no debe afectar la programabilidad ni romper la abstracción del patrón.

Suponga en el ejemplo del pipeline que procesa una secuencia de imágenes, que alguna etapa necesita descartar una cierta imagen de la secuencia. Para este caso se necesitarán acciones explícitas que indiquen cuando una actividad no debe dejar pasar la imagen a la etapa posterior. En este caso las actividades del pipeline tienen modo de interacción *explícito*.

3 SISTEMA ESQUELETAL: IMPLEMENTACIÓN

La necesidad de un sistema esqueletal que soporte la integración (anidamiento) y composición de los esqueletos es vital para componer programas paralelos más complejos que puedan ser expresados como una colección de actividades interactuantes. Aún con esta integración es casi imposible pensar que pueda resolver todos los problemas paralelos en forma eficiente,

es necesario un mecanismo que permita integrar estas construcciones de alto nivel con operaciones de comunicación y sincronización simples (Ej. Send, Receive, Barrier) y colectivas (Broadcast, Gather, Reduce). Para soportar estos mecanismos y evitar la introducción de nueva sintaxis, el sistema de programación esqueletal esta implementado como una librería de C, y hace uso de MPI para implementar los aspectos de comunicaciones. La librería MPI es altamente conocida por los programadores paralelos, y provee una excelente portabilidad.

Las características que nombramos en la sección anterior han sido incorporadas al sistema de programación esqueletal, permitiendo al usuario realizar una mejor configuración del esqueleto para resolver su programa paralelo. En una primera instancia de desarrollo se ha implementado el paradigma Divide y Vencerás.

3.1 Esqueleto Divide y Vencerás

El diseño y construcción del esqueleto Divide y Vencerás presentado, tiene sus sustentos en [8], y se ha modificado para soportar las características discutidas y otros rasgos que agregan mayor flexibilidad.

El esqueleto soporta un modo de anidamiento transitorio a través de invocaciones a otros esqueletos en cualquier momento de la ejecución del código interno de una actividad. Por ejemplo la actividad de división podría utilizar un esqueleto con interacciones centralizadas-centralizadas que lo resuelva. La opción de anidamiento persistente no es soportada aún por el esqueleto. Como ejemplos de sistemas que soportan ambos modos de anidamiento podemos nombrar: Skel [6], Kuchen's Skeleton Library [11] y Eden [3].

El modo de interacción es asociado con cada actividad en el esqueleto, instanciando el parámetro correspondiente en la llamada al mismo. Al contrario de otros esqueletos, donde debe definirse el modo de interacción por cada actividad, para el esqueleto $D\&V$ solo es necesario definir si este permitirá que las interacciones puedan realizarse en forma explícita. Para soportar las interacciones explicitas, el sistema provee una función *Interactuar()* que podrá ser invocada en cualquiera de las actividades (generalmente en las actividad de división y combinación).

Cuando el esqueleto es configurado en modo explícito su ejecución se asemeja a un código secuencial, en donde no existe comunicación entre procesos, la llamada a la función *Interactuar()* permitirá que las actividades socias interactúen en modo sincrónico. En este caso, el usuario del esqueleto debe llevar el control de la sincronización entre las actividades.

La definición del esqueleto es la siguiente:

```
void DC_Call(tipoDC Type,int Weight,mInteraction MI,TPF_trivial Itrivial,TPF_conquer Iconquer,
             TPF_divide Idivide,TPF_combine Icombine,TPF_secuencial Isecuencial,
             TypeN *In,int SizeBufferIn,int SizeDataTypeIn,TypeN *Out,int SizeBufferOut,
             int SizeDataTypeOut,MPI_Comm comm)
```

El prototipo del esqueleto $D\&V$ permite configurar un conjunto de parámetros, no sólo con el objetivo de proveer versatilidad en su aplicación a distintos problemas, sino también para que sea capaz de representar algunos de los conceptos esenciales mencionados anteriormente:

· **tipoDC Type**

Su valor determina la versión del algoritmo que se desea utilizar y la cual puede depender del tipo de problema específico a resolver. Es un tipo enumerado definido como:

```
typedef enum {DCC,DCH,DCED,DCCT) tipoDC;
```

1. DCC, Divide y Vencerás Clásico: en este algoritmo un sólo procesador dispone de los datos de entrada (entrada centralizada). Cuando el procesador divide el problema, comunicará parte de los subproblemas a procesadores libres y seguirá trabajando con el subproblema restante.

2. DCH, Divide y Vencerás Hipercúbico: en este tipo de algoritmo, el esqueleto mantiene una estructura recursiva y genera un árbol binario de grupos de procesos, cuyas hojas contienen un único proceso. En cada recursión, el grupo de procesos es dividido en dos subgrupos donde cada uno soluciona un subproblema. Esta configuración es ideal para algoritmos $D\&V$ binarios con interacciones RR.

3. $DCED$, Divide y Vencerás Completamente Paralelo (Divide and Conquer with Embarrassing Divisibility): este algoritmo pertenece a la clase de $D\&V$ en el cual no es necesario (o es mínima) la comunicación. En la práctica estos algoritmos son aquellos en el cual el problema puede ser tratado inmediatamente como dos o más subproblemas; en estos algoritmos no es necesario movimientos de datos extras en la etapa de división. Ejemplos de este tipo de algoritmo son el producto escalar y la multiplicación de matrices balanceadas.

4. $DCCT$, Divide y Vencerás con Combinación Trivial: estos algoritmos son aquellos en los que una vez que el problema es dividido a su máxima expresión, el mismo queda resuelto y su solución queda distribuida en todos los procesos de último nivel. En estos algoritmos la fase de división, no sólo divide los datos, sino que los modifica en busca del resultado. Como ejemplo de este tipo de algoritmo se puede mencionar el mergesort.

· **int Weight**

Este parámetro es también conocido como factor de ramificación (branching factor). Es el número de actividades (subproblemas) que se generarán en la fase de división. En cualquier caso $Weight$ debe ser mayor que 1. Para el tipo DCH, $Weight$ debe ser 2. En otro caso $Weight$ debe ser escogido teniendo en cuenta las características del problema y los procesadores disponibles.

Problemas como el quicksort tienen un factor de ramificación de 2, por más que el problema inicial sea dividido en 3 listas, sólo dos de ellas constituyen los subproblemas a resolver. Algunos algoritmos que trabajan sobre un espacio bidimensional (como el algoritmo $n-body$ sobre un plano) tienen un factor de ramificación igual a 4. La multiplicación de matrices de Strassen [12] tiene un factor de ramificación igual a 7.

En los casos en los cuales el factor de ramificación no es naturalmente mapeado sobre una arquitectura, alguna técnica de agrupación de procesos o balance de carga es necesaria. El esqueleto soporta un factor de ramificación constante. Una vez definido, es imposible cambiar este valor dinámicamente.

· **mInteraction MI**

Es un tipo enumerado definido como:

```
typedef enum {IMPL,EXPL) mInteraction;
```

Cada valor especifica el modo de interacción que se desea utilizar. El valor $IMPL$ indica que el esqueleto trabaja en modo implícito. En caso de configurarse en modo $EXPL$, las interacciones entre las actividades serán responsabilidad del programador.

· **TPF_trivial Itrivial**

Este parámetro es el puntero a la función $trivial()$ que indica sobre su retorno si se ha alcanzado el punto final de la recursión en donde no es posible seguir dividiendo los subproblemas. En tal caso, el esqueleto llama automáticamente a la función $conquer()$. Su prototipo es:

```
int trivial(TypeN *Input);
```

· **TPF_conquer Iconquer**

Apunta a la actividad encargada de resolver el caso base (trivial). El resultado debe ser almacenado en el parámetro $Output$. Su definición es la siguiente:

```
void conquer(TypeN *Input,TypeN *Output)
```

· **TPF_divide Idivide**

Representa el puntero a la actividad de división. Esta actividad divide los datos de entrada ($Input$) y genera el conjunto de subdatos a resolver. Es responsabilidad de la actividad de división generar una entrada (sub-problema) para cada ramificación, y asignar el espacio necesario. La estructura $IntraN$ contiene tantas entradas como número de ramificaciones se generen ($Weight$). El prototipo de la función es:

```
void divide(TypeN *Input, IntraN InputIntra)
```

· **TPF_combine Icombine**

Es la función encargada de combinar las soluciones parciales que se encuentran en $OutputIntra$, y de generar la solución general. Es responsabilidad de esta función asignar el resultado a $Output$. El prototipo de la función es:

```
void combine(TypeN *Output,IntraN *OutputIntra);
```

· **TPF_secuencial Isecuencial**

Es el puntero al algoritmo secuencial que (mejor) resuelve el problema en una arquitectura secuencial (arquitectura monoprocesador). El prototipo de la función es:

```
void secuencial(TypeN *Input,TypeN *Output)
```

· **TypeN *Input, int SizeBufferInput, int SizeDataTypeInput**

Este parámetro indica la estructura de entrada al esqueleto. La estructura $TypeN$ esta definida como un buffer genérico que puede representar cualquier tipo de dato elemental o definido por el usuario. El esqueleto necesita conocer el tamaño de su tipo de dato elemental ($SizeDataTypeInput$), y la cantidad de elementos ($SizeBufferInput$) requeridos. A continuación se muestra la definición de la estructura $TypeN$:

```
typedef struct
{
void * _Buffer;
long _Size;
}TypeN;
```

Para mayor facilidad en el uso de esta estructura se han implementado algunas de las operaciones más requeridas sobre ella. La función $DC\_Alloc()$ permite asignar espacio de memoria. Si $DC\_Alloc()$ no puede asignar la memoria requerida, asigna a $buff$ el valor $NULL$. El prototipo de la función es:

```
void DC_Alloc(TypeN *buff,int LongBuffer, int SizeDataType);
```

La función $DC\_Copy()$ permite copiar un buffer de datos. Antes de la invocación a $DC\_Copy()$, el usuario debe haber realizado la asignación pertinente sobre la estructura dst. En caso de no realizar la copia, asigna a dst el valor $NULL$. El prototito de la función es el siguiente:

```
void DC_Copy(TypeN *dst,TypeN *src);
```

· **TypeN *Output,int SizeBufferOutput, int SizeDataTypeOutput**

Estos tres últimos parámetros le indican al esqueleto el buffer de salida. Esta estructura también debe ser provista de un espacio de memoria asignada antes de llamar al esqueleto. El tamaño del tipo de dato elemental $SizeDataTypeOutput$ y la cantidad de elementos del buffer de salida $SizeBufferOutput$ deben ser especificados. No necesariamente un problema $D\&V$ debe tener el mismo tipo de dato elemental para el problema y la solución, ni tampoco el mismo tamaño de datos de entrada que de salida.

· **MPI_Comm comm**

Representa al comunicador de MPI. Permite al esqueleto delimitar cual es el grupo de procesadores involucrados en una comunicación, así también como la integración del esqueleto con código paralelo (Ad-Hoc) necesario para resolver un problema irregular que no se ajusta a ningún patrón soportado por el sistema.

3.2 Ejemplos

Para mostrar como el esqueleto resuelve distintos tipos de problemas Divide y Vencerás se han elegido ejemplos simples (”toys examples”). La idea de mostrar estos ejemplos es evidenciar como el esqueleto debidamente configurado, nos ayuda a resolver de forma práctica muchos problemas de la clase $D\&V$.

Reducción de Vectores El algoritmo de reducción de vectores es un ejemplo clásico que consiste en aplicar una operación asociativa y conmutativa a los valores correspondientes a cada una de las componentes de los vectores de entrada y de esa manera obtener un único vector resultado. Por ejemplo, dados dos vectores $A = (2, 4, 3)$ y $B = (6, 3, 5)$ y la operación suma $(+)$, el vector reducción es: $R = A + B = (2+6, 4+3, 3+5) = (8, 7, 8)$. Una solución eficiente usa el esqueleto configurado con los siguientes valores:

```
void DC_Call(DCED, 2,IMPL, &trivialRV, &conquerRV, NULL, &combineRV, &secuencialRV,
             &VectorInput,N, sizeof(vector), &VectorOutput, 1, sizeof(vector), Comm)
```

El primer parámetro indica que el esqueleto debe resolver un tipo de $D\&V$ completamente paralelo. Este tipo de $D\&V$ no necesita de una función de división. El esqueleto divide el conjunto de datos por el factor de ramificación hasta alcanzar el punto de parada ($\&trivialRV$), luego comienza su fase de conquista ($\&conquerRV$) y reducción ($\&combineRV$).

Otra configuración del esqueleto podría usar interacción explícita entre las actividades:

```
void DC_Call(DCED,2,EXPL,&trivialRV, &conquerRV, NULL, &combineRV, &secuencialRV,
        &VectorInput, N, sizeof(vector), &VectorOutput, 1, sizeof(vector), Comm)
```

En este caso el usuario debe realizar las comunicaciones necesarias de forma explícita en la actividad de combinación. Para realizar estas comunicaciones, el programador debe hacer uso de las funciones dispuestas por el sistema. Para el esqueleto Divide y Vencerás el sistema dispone de la función $Interactuar()$. La función se encarga de intercambiar el contenido de los bufferes de actividades socias.

```
void combine(TypeN *Output,IntraN OutputPartner)
{
 Interactuar(Output, OutputPartner, sizeof(Datatype))
 /* tareas de la combinación */
}
```

Quicksort Quicksort es un algoritmo Divide y Vencerás que ordena una secuencia de números, dividiéndola recursivamente en sub-secuencias más pequeñas [4]. El ordenamiento de las secuencias mas pequeñas representa dos subproblemas completamente independientes que pueden ser solucionados en paralelo.

```
DC_Call(DCCT, 2, IMPL, NULL, NULL, &divideqs, NULL, &secuencialqs, &VectorInput,
        N, sizeof(int), &VectorOutput, N, sizeof(int), comm)
```

Para la correcta instanciación del problema, se instancia el tipo $D\&V$ con combinación trivial. En este caso el esqueleto hace caso omiso de la función trivial, conquer y combine. La implementación del esqueleto resuelve el caso simple al llegar a un tamaño de entrada de datos de 1 elemento y luego se encarga de ir comunicando las soluciones parciales en el árbol y dejar la solución en $VectorOutput$.

FFT La Transformada Rápida de Fourier (FFT de Fourier Fast Transform) es uno de los métodos númericos más altamente usado en las ciencias e ingenierías, especialmente en el area de procesamiento de imagenes y señales, análisis espectrales, comunicación, teléfonos celulares y hasta sistemas de control digital. Existen varios algoritmos secuenciales que resuelven la FFT, uno de los primeros fue introducido por Cooley y Tukey (1966), ver [10] para más detalles.

```
DC_Call(DCH,2,IMPL,&Trivialfft,&conquerfft,&dividefft,&combinefft,&secuencialfft,&VectorInput,
        N, sizeof(Complex),&VectorOutput,N,sizeof(complex),comm);
```

Este ejemplo se adapta perfectamente al paradigma $D\&V$ con comunicaciones hipercúbicas. Por otro lado, esta aplicación permite testear el esqueleto en su máxima expresión y mostrar como funcionan todas sus componentes. La actividad $dividefft$ se encarga de dividir la secuencia de entrada ($VectorInput$) en los componentes pares e impares hasta llegar a secuencias de 1 elemento, donde la FFT de un elemento es el mismo elemento ($trivialfft$ y $conquerfft$). Por último $combinefft$ realiza las computaciones necesarias y combina las secuencias transformadas resultantes en $VectorOutput$.

4 CONCLUSIONES Y TRABAJO FUTURO

En este trabajo, se revisaron algunos conceptos fundamentales de diseño que deben soportar los sistemas esqueletales (conjunto de constructores de alto nivel) si se desea obtener un entorno de programación esqueletal flexible que ayude al programador paralelo a resolver una amplia gama de problemas. También se ha introducido y utilizado una terminología fácil de comprender y a la vez ágil para definir estos conceptos y establecer su relación con un sistema de programación esqueletal. El soporte de anidamiento es esencial en la programación estructurada para poder combinar e integrar los distintos esqueletos y encarar soluciones a problemas irregulares y complejos. Un sistema esqueletal que soporte ambos tipos de anidamiento ofrece al usuario mayor flexibilidad al momento de diseñar una solución. Si bien, en la actualidad, el sistema implementado soporta únicamente el anidamiento transitorio, se ha podido observar como esta caracterítica es útil para representar problemas variados. El modo de interacción es otro aspecto que debe tenerse en cuenta si lo que se requiere es flexibilidad. El modo de interacción explícito elimina las restricciones temporales entre las actividades, pero incrementa la responsabilidad del usuario. Nuestro sistema soporta ambos modos de interacción e introduce una función de comunicación colectiva definida para el esqueleto divide y vencerás.

Si bien el esqueleto puede resultar complejo por su número de parámetros, se ha mostrado, por medio de la instanciación de distintos ejemplos clásicos, que la interfase es conceptualmente manejable y que su correcta configuración facilita la resolución de problemas típicos pertenecientes a la clase de problemas, en este caso, $D\&V$.

En la actualidad se esta trabajando en la generación de otros esqueletos, en el soporte de características avanzadas (balance de carga, anidamiento persistente, etc), la portabilidad y otimización de esqueletos en diferentes arquitecturas (incluidos Multiclusters y Grid). Todos estos temas son relevates si queremos que un sistema de programación esqueletal alcance los objetivos de programabilidad, portabilidad y performance, y demuestre obtener beneficios con su uso.

AGRADECIMIENTOS

Los autores deseamos agradecer a la Universidad Nacional de San Luis, la ANPCYT y el CONICET por su contínuo soporte en el desarrollo de nuestras investigaciones.

BIBLIOGRAFÍA

[1] Message passing interface forum. available at http://www.mpi-forum.org/docs/docs.html.

[2] M. I. Cole. A. Benoit. *Two Fundamental Concepts in Skelletal Parallel Programming.* ICCS 2005, LNCS 3515, 2005.

[3] Loogen R. Ortega-Mall´en Y. PeÜna R. Breitinger, S. *Eden: Language Definition and Operational Semantics.* Technical Report 10, Philipps-University of Marburg, 1996.

[4] Hoare C.A.R. *Algorithm 64: Quicksort*. Communications of the ACM, 4, p. 321, 1961.

[5] M. I. Cole. *Algorithmic Skeletons: Structured Management of Parallel Computation. Research Monographs in Parallel and Distributed Computing.* Pitman, London, UK., 1989.

[6] M. I. Cole. *eSkel: The edinburgh Skeleton library Version 2.0*. Draft API reference manual. Internal Paper, School of Informatics, University of Edinburgh, 2003.

[7] C. Rodriguez León F. Piccoli, M. Printista. *Dynamic Hypercubic Parallel Computations*. Proceeding (466) Parallel and Distributed Computing and Systems, 2005.

[8] M. Printista F.D. Saez, R. Gallard. *Paradigms of Parallel Programming*. Workshop de Investigadores en Ciencias de la Computación (WICC 2003), Tandil, Argentina, May 2003.

[9] M. Printista J.G. Zanabria, F. Piccoli. *Hypercubic Comunications in MPI*. Tesis submited for UNSL, 2005.

[10] Cooley J.W. and Tukey J.W. *An algorithm for machine calculation of complex Fourier series*. 1966.

[11] H. Kuchen. *A skeleton library*. Proceedings of the 8th International Euro-Par Conference on Parallel Processing, Springer-Verlag Pag. 620-629, 2002.

[12] Strassen V. *Gaussian elimination is not optimal*. Numerische Mathematik., 1969.

Control Evaluation in a LVoD System Based on a Peer-to-Peer Multicast Scheme*

Rodrigo Godoi, Xiaoyuan Y. Xu, Porfidio Hernández and Emilio Luque
Computer Architecture and Operating Systems Department.
Universitat Autònoma de Barcelona, UAB.
Edifici Q, Barcelona 08193, Spain.
{rodrigo, xiao}@aomail.uab.es, {porfidio.hernandez, emilio.luque}@uab.es

Abstract

Providing Quality of Service (QoS) in video on demand systems (VoD) is a challenging problem. In this paper, we analyse the fault tolerance on a P2P multicast delivery scheme, called *P*atch *C*ollaboration *M*anager / *M*ulticast *C*hannel *D*istributed *B*ranching (*PCM/MCDB*) [13]. This scheme decentralizes the delivery process between clients and scales the VoD server performance. PCM/MCDB synchronizes a group of clients in order to create local network channels to replace on-going multicast channels from the VoD server. Using the P2P paradigm supposes facing the challenge of how often peers connect and disconnect from the system. To address this problem, a centralized mechanism is able to replace the failed client. We evaluate the failure management process of the centralized scheme in terms of the overhead injected into the network and analyse the applicability of a distributed approach to managing the process. Analytical models are developed for centralized and distributed approaches. Their behaviour are compared in order to evaluate whether the distributed scheme can improve the fault management process, in terms of reducing server load and generating better scalability.

Keywords: VoD, Multicast, P2P, Fault-tolerance.

Resumen

Proporcionar Calidad de Servicio (QoS) en sistemas de Vídeo bajo Demanda (VoD) es un problema desafiador. En este artículo, analizamos la tolerancia a fallos en un esquema de envío de informaciones, basado en comunicaciones *multicast* y colaboraciones P2P, denominado PCM/MCDB [13]. El esquema descentraliza el proceso de envío de información entre los clientes y escala las prestaciones del servidor de VoD. PCM/MCDB sincroniza un grupo de clientes con objeto de crear canales de redes locales para reemplazar canales *multicast* en curso del servidor. La aplicación del paradigma P2P supone cómo afrontar el problema de la conexión y desconexión de clientes del sistema. Para resolver este problema, un mecanismo centralizado es capaz de reemplazar el cliente fallido. En el trabajo evaluamos el proceso de gestión de fallos del esquema centralizado en términos del flujo de informaciones insertado en la red y analizamos la aplicabilidad de un esquema distribuido para el proceso de gestión. Modelos analíticos son desarrollados para las aproximaciones centralizada y distribuida. Sus comportamientos son comparados con objeto de evaluar si un esquema distribuido puede mejorar el proceso de gestión de fallos desde el punto de vista de reducir la carga del servidor y proporcionar mejor escalabilidad.

Palabras claves: VoD, Multicast, P2P, Tolerancia a fallos.

* This work was supported by the MEyC under contract TIN 2004-03388.

1. INTRODUCTION

Recent advances provide multicast scheme application on real networks. This allows clients to share delivery channels and decrease the server and network resource requirements. The patching multicast policy [01] [02], for example, dynamically assigns clients to join on-going multicast channels and patches the missing portion of video with a unicast channel. The disadvantage of a multicast scheme, compared with unicast, is the complexity of implementing interactive operations, because there is not a dedicated channel per client.

Most recently, the peer-to-peer (P2P) paradigm has been proposed to decentralize the delivery process to all clients, achieving system scalability beyond the physical limitations of VoD servers. In [03] [04], the authors propose the Chaining delivery policy to link clients in a delivery chain. Even though P2P policies achieve high resource requirement reduction in the server, the schemes' applicability in a true-VoD system has been questioned due to client failure problems.

The P2Cast P2P delivery scheme [09] creates a delivery multicast tree and is able to combine the Patching and Chaining policies. P2VoD [07] introduce the concept of generation, which groups a set of clients in the information-propagation process. P2VoD and P2Cast present P2P VoD systems with fault-tolerance mechanisms based on the recursive reconstruction of the delivery tree. Neither of them evaluates the cost involved in the failure management process, which is very important, since the system presents restrictions in order to maintain the QoS.

In [05] [06], we proposed a P2P delivery policy that was able to synchronize a set of clients to collaborate with the server. However, [05] [06] does not provide a client-failure recovery mechanism. In [08] the authors present a study of the P2P paradigm applied to file sharing, where they show the significant amount of heterogeneity in this kind of system. The observations lead to a set of problems that we must also take in account for VoD systems.

In this paper we propose a failure management process, based on a P2P multicast delivery scheme, named *PCM/MCDB*. This scheme allows clients to collaborate with the server in a distributed way to send video information using multicast channels. The scheme is designed as two separate P2P policies. The first policy (PCM) creates multicast channels from the server to send video information and indicates collaborative clients for patching the missing portion of video. In the second policy (MCDB), we introduce the idea of a multicast channel branching where a group of clients is synchronized to generate local network multicast channels (branches).

Unlike traditional P2P schemes, client failures in the new delivery scheme do not immediately affect the QoS. We developed an advanced client failure detection mechanism in which each client in a collaboration group monitors neighbouring clients. The failure mechanism is able to detect a client failure before delivery disruptions occur. Once a client failure is detected, a centralized failure recovery policy is triggered in the server to replace the failed client with another, providing continuous video playback without glitches.

In our study, the proposed centralized scheme is evaluated using an analytical model developed according to server, network and client parameters. A distributed scheme is also analysed in order to improve the performance of the VoD system in terms of server resource requirements and scalability. Our analysis is based on the overhead introduced into the network by the failure management process.

The remainder of this paper is structured as follows. Section 2 presents the key ideas behind our delivery scheme. The Failure Management Process, using centralized and distributed approaches, is analyzed in Section 3. In Section 4, the developed analytical models are presented. Performance evaluation is shown in Section 5. In Section 6, we indicate the main conclusions of our results and future studies.

2. P2P MULTICAST DELIVERY SCHEME

In this section we introduce the environment considered and present the delivery policies adopted. The control analysis is developed taking multicast islands into account. This means networks have routers with IP Multicast capability. Thus, the models developed are only applied in this environment in a first approach. The entire VoD system is composed of distinct multicast islands and the communication between the islands is through border routers using unicast channels.

2.1. Overview of P2P VoD Architecture

In a video service, video information is sent by the server through the network to clients. There are 3 main components that implement the VoD architecture: server, clients and network. The server design defines the data organization strategy, data retrieving process from the storage system and the data delivery process to the network interface. All these modules have to be designed in order to satisfy the soft real-time requirements of the video delivery process.
The client receives, decodes and displays the video information. Throughout the process, the client design includes buffers that temporarily cache received information for 3 purposes:
1) A portion of this buffer is used to achieve smooth playback. We call this portion the jitter buffer. The size of this portion is invariable and is mainly dependant on the video format and variations in network bandwidth. 2) The client caches video information from the delivery channels (delivery buffer). The size of the delivery buffer changes according to the delivery policy. 3) All the client buffer that is not used for the previous 2 purposes will be utilized in the client collaboration. We call this portion of buffer “collaborative buffer” and it is able to cache video information for sending to another client.
Clients are connected to the VoD server through the network. In our design, we assume that the network is segmented and each client is able to maintain independent communication with other clients. We also assume that the local client is able to deliver video information to the local network using the multicast technique.
Video information is assumed to be encoded with a Constant Bit-Rate (CBR). The video information is delivered in network packets and the packet size is invariable. We call a network block a video block. We enumerate the blocks of a video from 1 to L, L being the size of a video in video blocks.

2.2. PCM and MCDB Policies

The delivery scheme decides how the video information is sent to clients. Our delivery scheme is designed based on two policies (Figure 1): (a) *Patch Collaboration Manager* (PCM) and (b) *Multicast Channel Distributed Branching* (MCDB).
The objective of PCM is to create multicast channels to service groups of clients, and allows clients to collaborate with the server to deliver portions of video in the admission process. With PC**M,** clients receive video information from both a multicast and unicast channel. The multicast channels are created by the server, whereas the unicast channels could be created either by the server or the clients. Multicast channels deliver every block of a video while unicast channels only send a portion of a video. We call the multicast channel a Complete Stream and the unicast channel a Patch Stream (Figure 1 a).

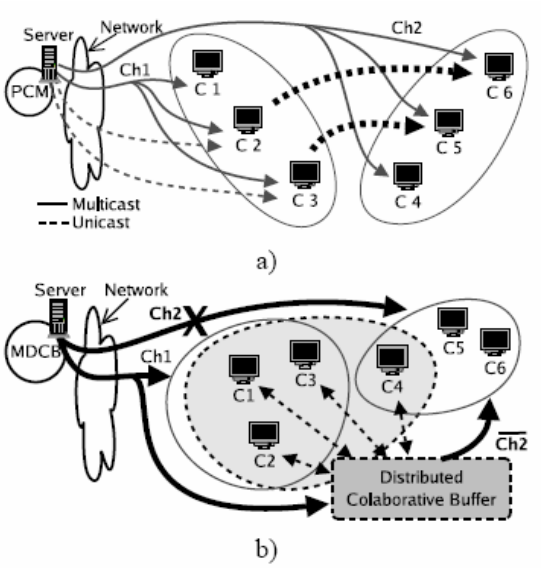


Figure 1. Delivery Scheme: a) PCM Collaboration. b) MCDB Collaboration.

The objective of MCDB, however, is to eliminate multicast channels so as to reduce server load. The policy replaces an on-going multicast channel with a local multicast channel. A group of collaborative clients is synchronized to form a Distributed Collaborative Buffer. Clients of this group use their buffers to cache video blocks from another multicasting channel. The cached blocks are delivered by the collaborative clients in order to generate the local multicast channel. When a multicast channel is replaced by one generated with collaborators clients, we call the new channel a branch channel (Figure 1 b).

3. NODE FAILURE MANAGEMENT PROCESS

In VoD systems, failures can be caused by a network failure, a client machine crash or even VCR operations. Furthermore, in P2P based systems, peers come and leave freely, so a client departure can be faced as a failure for the system. In such a situation, the client stops sending video information that can degrade the QoS. To address the problem of a failed client, we use a Failure Management Process based on three components: failure detection, recovery and maintenance of the system's information coherence.

3.1. Failure Detection

Client failure detection supposes that a collaborator suddenly leaves the system. The MCDB associates each client with 2 neighbouring clients in accordance with the client position in the distributed circular buffer. For instance, in Figure 2, client C2 has clients C1 and C3 as neighbouring clients. Each client periodically receives 2 synchronization messages from its neighbours. The messages notify the state of the neighbours and, if one of the neighbours has failed, the client sends a control message to the element responsible for starting the recovery process. This detection mechanism is able to detect a failure in advance because only the client, in the collaborative group, that is sending the video information affects the quality of the branch-channel. In Figure 2, a failure of client C2 does not affect the quality of branch-channel until client C3 finishes delivering block 13 and 14. This approach to failure detection is distributed, once every node receives messages from other nodes.

3.2. Failure Recovery

In PCM/MCDB policy, the recovery process is centralized in the server and is triggered when a client failure is detected. The centralized approach supposes a simpler design and can represent an adequate and efficient solution for a range of multicast applications, since the server is responsible for all the processes and their steps. However, a centralized architecture has obvious implications, such as server load or the fact that a single controller represent a single point to manage all nodes' failure operations, and if this crashes, the whole fault tolerance scheme is lost. To address these problems we evaluate the centralized scheme and propose a distributed one and analyse its performance.
The recovery process defines different recovery actions according to the state of a failed collaborative client. In the MCDB P2P delivery process, a collaborative client can be in 4 states: 1) one client in the group caches the video information from other multicast channel. 2) a client could be waiting to start the delivery process. 3) One client is delivering video information to the branch-channel. 4) Clients could be waiting to start the caching process. In Figure 2 a), C1 is caching, C2 is waiting to start delivering, C3 is delivering and C4 is waiting to start caching.

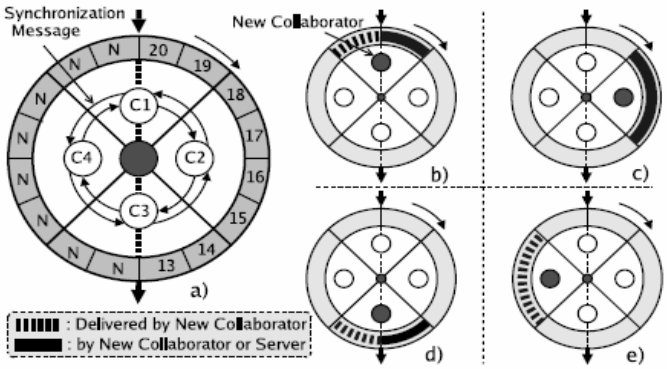


Figure 2. Client Failure Recovery: a) Advance Failure Detection. b) Recovery Process of State 1. c) Recovery Process of State 2. d) Recovery Process of State 3. e) Recovery Process of State 4.

State 1 Caching: The failure recovery policy tries to find a new collaborative client to replace the failed client. The new collaborator continues with the cache process of the failed client. The video blocks that are already cached by the failed client will be delivered by the new collaborator, if the collaborator has these video blocks. For example, in Figure 2 b), C1 fails and the new collaborator will cache blocks 21 and 22. Block 19 and 20 will be sent by the server or by the new collaborator.
State 2 Waiting for delivering: The recovery action for this state is quite similar to the case for state 1. However, the new collaborator or server has to send all the video blocks that are cached by the failed client (Figure 2 c).
State 3 Delivering: In this case, the recovery process finds a collaborator to replace the failed client, just **as** in state 1. The new collaborator does not need to have the cached information of the failed client, but has to have the same collaborative buffer capacity. In case the new collaborator does not have video information, the server assumes the delivery process of the failed client temporarily. For example, if C3 in Figure 2 d) fails, and the recovery process is unable to find a new collaborator that has blocks 13 and 14, the server will continue sending blocks 13 and 14.
State 4 Waiting for Caching: The failed client has no useful information in its buffer, so the recovery process only needs to find a new collaborator to replace it (Figure 2 e).
A client failure in states 1, 2 and 4 does not immediately affect the branch channel's delivery process. Thus, the recovery process could take place with a certain delay without affecting the

delivery process. However, if the failed client is in state 3, the delivery process will be immediately disrupted. Since the client failure detection mechanism needs a period of time to trigger the recovery policy, clients of the disrupted channel will not receive any information before the recovery process. In order to avoid glitches, we delay the playback for a short period to create a cushion buffer. This cushion buffer provides video information until the end of the recovery process.
In PCM, the collaborative client is always sending video blocks, so the collaborator only has a state like the 3 in MCDB.
Beyond the margin that the cushion buffer gives the system, in the case of the MCDB policy, we have a more flexible situation in 75% of cases, because the clients' disruption does not immediately affect the transmission (states 1, 2 and 4). However, in order to keep the system working with QoS, we assume the critical cases where an efficient solution is required, which occurs when a node is sending video blocks (state 3 and PCM policy).
In a general sense the recovery stage depends on the system's failure frequency (f_e), the message changing to trigger the recovery process and the message changing between routers in order to maintain or rearrange the distribution tree.

3.3. Maintenance of the System Information

This stage of the Failure Management Process is not connected directly to the other two, in the sense of not being part of the logical sequence, but it is as important, because it has a great influence on the recovery process.
The maintenance of information coherence means that the system must keep the state of the nodes with certain precision, whether they are available or not and all the data needed to perform a set of possible collaborators.
Out of date information, originated by a bad maintenance process, can lead to a wrong collaborator selection, which involves an answer refused by the indicated peer. Thus, a new search for collaboration must be triggered and the time to solve a node failure will increase.
A centralized scheme to maintain node information up to date supposes the server receiving information messages from all active clients on a receive frequency (f_{CI}). In the other hand, a similar process is needed by the routers for the multicast distribution trees, that is, each one sending and receiving messages, in a distributed way, to maintain or update the multicast tree.

4. COSTS OF FAILURE MANAGEMENT PROCESS

To improve the performance of the Failure Management Process, we propose the analysis of two different approaches for the process: a centralized and a distributed one. In order to evaluate both approaches, we analysed the volume of control messages injected into the network to achieve all the three phases, Detection, Recovery and Maintenance. This *overhead* metric represents the cost of each scheme. In this section, we formalize the overhead for each part of the failure management process. For convenience, the parameters used in the analysis are defined in table 1.
The PCM/MCDB already assumes the Detection phase in a distributed approach, so, we define its cost, in function of the Overhead. It is determined for the send frequency of *heart beat* messages and for the number of messages needed in this process, which must be changed for all nodes in every existing group. The cost of detection is therefore given by:

$$C_{\det ection} = f_{HB} \cdot \beta \cdot \sum_{i=1}^{G} N_{C\_g(i)} \qquad (1)$$

The Recovery and Maintenance phases are proposed in a centralized way for PCM/MCDB. Therefore, in the following sections, we analyse these centralized approaches and their associated costs and present the decentralized one, with its respective costs.

$C_{overhead}$	Number of messages injected into the network.
$C_{detection}$	Number of messages injected into the network in the detection phase.
$C_{recovery}$	Number of messages injected into the network in the recovery phase.
C_{maint}	Number of messages injected into the network in the maintenance phase.
N_C	Number of active clients in the system.
H	Number of clients that trigger a recovery process.
G	Total Number of multicast groups.
$HOPS_{g(i)}$	Number of hops for each multicast group G.
p_s	Probability to find a collaborator.
f_{HB}	*Heart beat* messages frequency.
f_e	Faults occurrence frequency.
f_{CI}	Client communication messages frequency.
f_{TI}	Router communication messages frequency.
β	Number of messages required for the detection protocol.
σ	Number of messages required between clients for the recovery protocol.
γ	Number of messages required between routers for the recovery protocol.
ω	Number of messages required between clients for the maintenance protocol.
α	Number of messages required between routers for the maintenance protocol.

Table 1. Parameters used in the analysis

4.1. Recovery

The Recovery process depends on client communication to trigger the process. Messages are sent to start the process and an answer is received, so the new collaborator's connection can be performed. On the other hand, a communication between routers that implements the IP Multicast is also necessary to arrange the distribution tree. This process is inherently distributed, since it is a question of routers, but communication between clients can be taken as a centralized or distributed approach.
Centralized: The centralized scheme supposes all faults, which trigger a Recovery process, make clients contact a central server. This server is responsible for performing a search based on clients' information. The search should select the most suitable collaborator to substitute the failed one. After selecting an adequate candidate, the server contacts the nodes implied in the Recovery and the new collaborator, to perform the link. So, the Overhead cost, considering router communication is given by:

$$C_{recovery} = f_{e.} \cdot \left[\sigma \cdot H + \sum_{i=1}^{G} \gamma \cdot Hops_{g(i)} \right] \quad (2)$$

Distributed: The distributed scheme assumes the triggered Recovery process is managed autonomously by its own nodes. We define a Manager Node per Multicast group, which is responsible for keeping information about the group members.
The Manager Node is a client that has the responsibility for managing a Multicast group, because it has full member information. The selection of this node is performed based on its history in the system and its capabilities, such as buffer size, process capacity and available bandwidth. A hierarchy is established in the Multicast group, in order to enable attribution of the function of Manager for another node in a set of nodes, in case the Manager fails.

On a first attempt, the Manager Node receives the recovery query and searches a substitute node in its clients' group information list. This phase verifies the existence of a candidate with the necessary characteristics to substitute the failed peer. If there is a node capable in the group, the linking process is performed; otherwise, the Manager Node contacts the Manager Node of another group, and asks for a qualified candidate to replace the failed peer. This process is repeated for all groups, until a new collaborator is found, always respecting a threshold time in order to maintain the QoS. After selecting an adequate candidate, the contact between the nodes that query for Recovery and the new collaborator is established, to perform the join. Thus, the Overhead cost for this case, considering router communication, is given by:

$$C_{rec.} = f_{e.} \cdot \left(\frac{H \cdot (1 + \sigma \cdot p_s)}{p_s} + \sum_{i=1}^{G} \gamma \cdot Hops_{g(i)} \right) \qquad (3)$$

4.2. Maintenance

The Maintenance means that the system must keep node information (content, buffer size, bandwidth, etc.). The exactness of this information determines how successful the recovery process will be.

To provide the set of possible collaborators with up-to-date information, messages are exchanged between the clients or between the clients and the server, depending on the scheme adopted, centralized or distributed. We evaluate these two approaches below. The Maintenance process also needs communication between the routers that implement the IP Multicast in order to maintain or rearrange the Multicast groups. This process is inherently distributed, given that it is a question of routers.

Centralized: The centralized scheme consists of clients sending periodic messages to a central server to inform about their characteristics. The server analyses the information and creates lists with a set of possible collaborators. Therefore, the Overhead cost, considering router communication, is given by:

$$C_{ma\,\mathrm{int}} = f_{CI} \cdot \omega \cdot N_C + f_{TI} \sum_{i=1}^{G} \alpha \cdot Hops_{g(i)} \qquad (4)$$

Distributed: The distributed scheme supposes that clients inside a Multicast group exchange messages periodically to inform about their characteristics. In this case, there is no central point that contains all the node information. All peers in a Multicast group send messages to the Manager Node, who analyses the information and creates lists with a set of possible collaborators. The process is the same for all groups. Thus, the Overhead cost for this case, considering router communication, is given by:

$$C_{ma\,\mathrm{int}} = f_{CI} \cdot \left(\sum_{i=1}^{G} \omega \cdot N_{C\_g(i)} + G^2 \right) + f_{TI} \cdot \sum_{i=1}^{G} \alpha \cdot Hops_{g(i)} \qquad (5)$$

4.3. Unicast Cost Model

In order to evaluate the influence of the transmission scheme, we developed a model that represents the cost on the three process' phases, considering the unicast transmission scheme.

We adopted the centralized failure management process as background. Like the IP Multicast case, we define an analytic model to represent the cost of a centralized failure management process in a unicast environment. So, the expression for the cost is given by:

$$C_{overhead} = (f_{HB} \cdot N_C \cdot \beta) + (f_{e.} \cdot \sigma \cdot H) + [(f_{CI} \cdot \omega \cdot N_C) + (f_{TI} \cdot \alpha \cdot L_u)] \quad (6)$$

The parameters considered for the model are the same that was used to modelling the failure management process in the multicast environment. The difference between the multicast and unicast models is on the recovery and maintenance phases. On the unicast model, there are no groups; therefore there is no need to restructure the distribution tree when a failure occurs. On the maintenance phase, it's not necessary keep groups' state, but routes are up to date dynamically, based on system's characteristics. The unicast scheme creates a point-to-point communication channel, in which the information flows. In this way, the routers must periodically change messages and process the calculations of the routing algorithms. In our work, we calculate a mean route (L_u) based on the adopted topology, showed in figure 3 and given by:

$$L_u = \frac{\sum_{i=0}^{M-1} (M-i) \cdot 2^{(M-i)}}{\sum_{i=0}^{M-1} 2^{(M-i)}} \quad (7)$$

5. PERFORMANCE EVALUATION

The developed models are evaluated adopting a binary tree router topology. This topology is shown in Figure 3 and has seven levels ($M(i) = [1, 7]$). Values are attributed to the parameters on the centralized and the distributed schemes.

Each tree level has $2^{M(i)}$ routers, therefore the total number of routers in this topology is 254. We consider that each hop has an associate network that is limited to connect a maximum of 120 clients. The number of possible active clients in the system is:

$$N_C = \left(\sum_{i=0}^{M(7)-1} 2^{(M(7)-i)} \right) \cdot 120 = 30.480 \quad clients \quad (8)$$

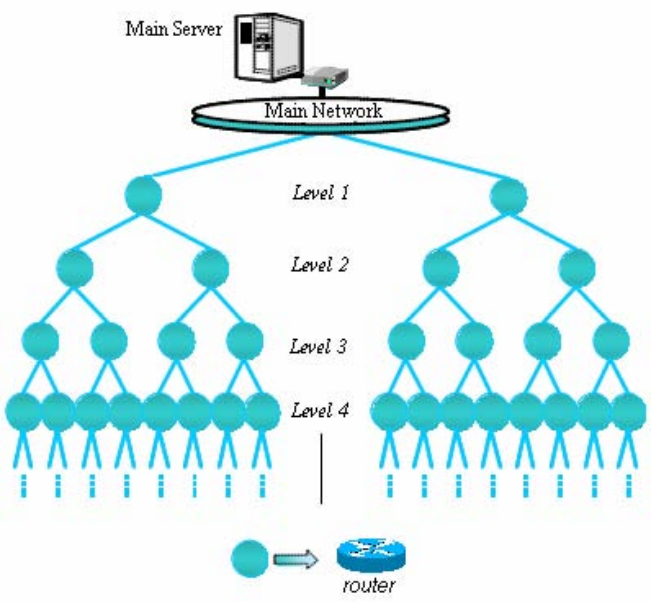


Figure 3. Binary Tree Topology

A single recovery request is considered (H = 1) and the probability of finding a collaborator (ps) was varied between 5%-95%. We verified that this parameter has little influence on the total messages amount, because the number of messages necessary in this communication is much lower than the number involved in the other phases of the process.
We are considering the PIM-SM protocol for the multicast implementation in the routers, because recently it is one of the most used in IP multicast. The number of messages in the detection, recovery and maintenance protocols were assumed as one (γ, ω, α) and two (β, σ). The frequency of sending heart beat messages (f_{HB}) is 1 every 5 seconds. We consider that the status messages (f_{CI} and f_{TI}) are sent with a frequency of 1 every 15 seconds [10] [11] [12].
We observe the behaviour of the centralized and the distributed approaches for three different parameters: the number of multicast groups, the quantity of clients in the system and the frequency of errors occurrence. The evaluation is made in the sense to measure the cost increment that the distributed scheme presents when it's compared to the centralized cost. This incremental cost is represented like a percentage, how defines the following expression:

$$\Delta = \frac{(C_{dist.} - C_{cent.})}{C_{cent.}} \cdot 100 \qquad (9)$$

The number of multicast groups is varied between 20 and 200 groups. The others models' parameters are fixed, such as to all next evaluations done. A multicast group can contain clients that are visualizing the same video and that arrived in the time interval necessary to join in a multicast channel. A multicast group also can be P2P collaboration groups, which share resources between clients. In figure 4 we can observe that the amount of messages grows with the number of the multicast groups in the system. The difference Δ, increases as the number of groups grows. This is caused because the distributed scheme for failure management, considers the communication of the Managers Nodes. In the maintenance phase there is a term G^2 that represents the communication between groups. The increasing Δ is caused by the decentralization policy adopted, that supposes groups communications, nevertheless this doesn't means that a distributed scheme is not scalable, how we analyse in the following evaluations.
In order to evaluate the scalability of the system when the quantity of clients grows, we vary this parameter and observe the behaviour of the message cost. In figure 5 is possible to verify that Δ decreases as the number of clients grows in the system. This diminution occurs because the communication between groups has less importance when compared with the amount of messages originated for the growing quantity of clients. It means, for a low number of clients dispersed in multicast groups the communication between Managers Nodes has major importance, nevertheless when the groups are more dense, the messages generated for the clients assumes more importance, so the difference Δ between the centralized and the distributed scheme decreases.
In LVoD systems that uses the P2P paradigm, the clients connect and disconnect with a certain frequency. The figure 6 shows the influence of the increment in the failure frequency. Each failure triggers a recovery process, so the cost increases. The difference Δ among the centralized and distributed schemes grows with the failure frequency. Although, in systems with characteristics like LVoD system, members join to view a specific content, which in general is not short, so the lifetime of a client in the system can be considered around 300 seconds, in the worst case [11]. In the case of 1 failure every 5 minutes the Δ is 5.7%, what doesn't represents a big overhead.
In order to evaluate the influence of the transmission schem, multicast and unicast were compared. The figure 7 represents the behaviour of the cost, in a centralized scheme of failure management, considering multicast and unicast transmissions. This analysis can show the influence of the communication between routers. The multicast diffusion requires routers' communications in order to maintain and construct distribution trees, in the other hand routers using unicast only needs

change messages periodically in order to up date the routing table. The increment in the number of clients in the system leads to a diminution in the Δ among multicast and unicast. This behaviour occurs because when the number of users grows, the messages interchanged by the routers have less importance if compared with the amount of messages generated by the clients.
These results shows that a distributed control scheme causes an increment in the network load, which can be considered acceptable in some cases, since the server is free of fault control, saving system resources that can be applied to any other function and the system has no single point for managing failures.

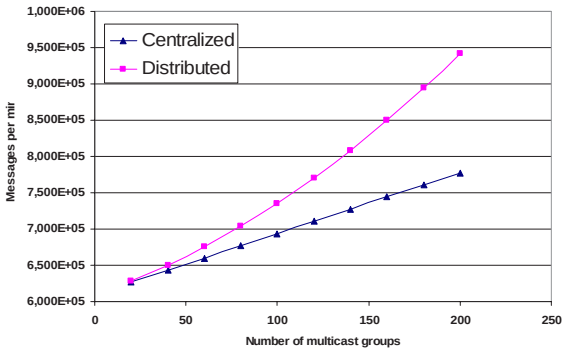


Figure 4. Influence of the number of multicast groups

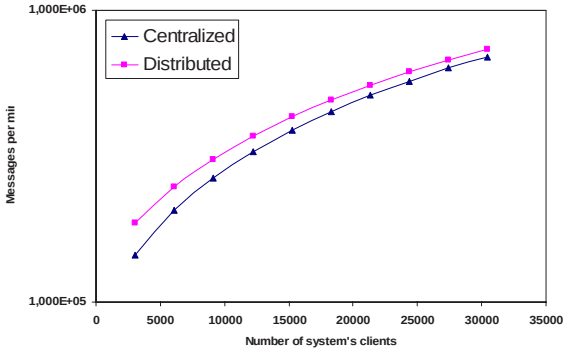


Figure 5. Influence of the number of clients

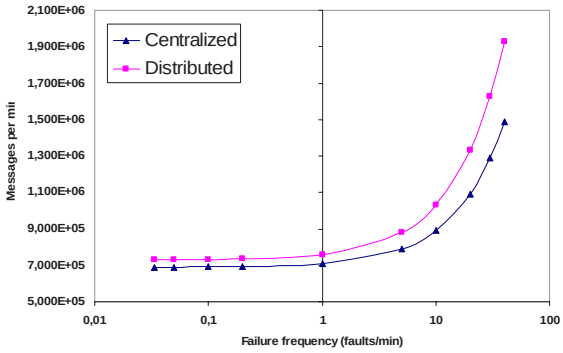


Figure 6. Influence of the failure frequency

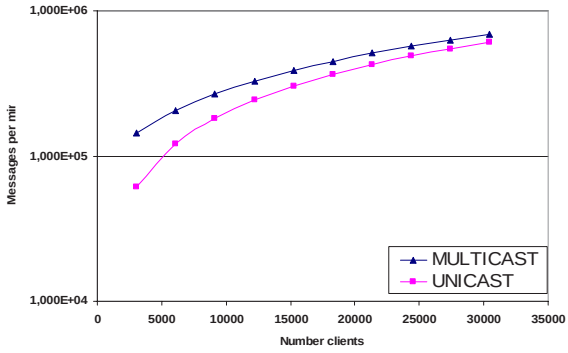


Figure 7. Multicast vs. Unicast

6. CONCLUSIONS

We analysed the cost of the failure management process, based on the volume of control messages, in the PCM/MCDB P2P multicast scheme. Analytic models were developed to represent the behaviour of centralized and distributed schemes, and to represent multicast and unicast communications. A system topology was defined in order to evaluate both schemes.
For large systems with many hops and clients, the distributed approach presents an inherent increase in messages number. Nevertheless, this increment could be assumed since the distributed approach saves system resource, frees the server of control load and creates multiple points to manage failures. Its applicability depends of tuning some parameters, like number of multicast groups, number of hops in the groups, or even defining others communications' protocols for a distributed failure management process.
Thus, a distributed control scheme can present many advantages, including a more feasible system, more scalability and resource saving, in exchange for some increase in the network traffic.

We have started several future research projects. First, our objective is to implement the analytic models in a simulator and compare the results. More research will be needed in order to evaluate the control schemes, and find the most suitable. Finally, we are studying the application of a control structure, composed for policies and maybe for dedicated elements to provide LVoD systems.

REFERENCES

[1] Cai, Y., Tavanapong, W., Hua, K. A. Enhancing patching performance through double patching. Proceeding of 9th Intl Conf. On distributed Multimedia Systems, 2003

[2] Hua, K. A., Cai, Y., Sheu, S. Patching: A multicast technique for true video-on-demand services. ACM Multimedia Conf, 1998.

[3] Hua, K. A., Sheu, S., Wang, J. Z. Earthworm: A network memory management technique for large-scale distributed multimedia applications. Proceedings of the INFOCOM '97, 1997.

[4] Jin, S., Bestavros, A. Cache-and-relay streaming media delivery for asynchronous clients. Proceeding of NGC'02, 2002.

[5] Yang, X. Y., Hernández, P., Cores, F., Ripoll A., Suppi, R., Luque, E. Distributed P2P Merging Policy to Decentralize the Multicasting Delivery. Proceeding of 31st EuroMicro Conference, 2005.

[6] Yang, X. Y., Hernández, P., Cores, F., Ripoll A., Suppi, R., Luque, E. Dynamic distributed collaborative merging policy to optimize the multicasting delivery scheme. Euro-Par, 2005.

[7] Do, T., Hua, K., Tantaoui, M. P2vod: providing fault tolerant video-on-demand streaming in peer-to-peer environment. Communications. IEEE International Conference, 2004.

[8] Saroiu, S., Gummadi, P. K., Gribble, S. D. A Measurement Study of Peer-to-Peer File Sharing Systems. Proceedings of Multimedia Computing and Networking, 2002.

[9] Guo, Y., Suh, K., Kurose, J., Towsley D. P2Cast: P2P Patching Scheme for VoD Service. Computer Science Technical Report 02-34, 2002.

[10] Wang, X., Yu, C., Schulzrinne, H., Stirpe, P., Wu, W. IP Multicast fault recovery in PIM over OSPF. Proceedings of ACM SIGMETRICS, 2000.

[11] Silverston, T., Fourmaux, O. Measuring P2P IPTV Systems. ACM NOSSDAV, 2007.

[12] Tarik C., S. Gjessing and O. Kure. Tree Recovery in PIM Sparse Mode. In Telecommunication Systems 19:3,4, 443–460, 2002.

[13] Yang, X. Un Sistema de Vídeo-bajo-Demanda a gran escala basado en la Arquitectura P2P con Comunicaciones por Multidifusión. PhD thesis, Universitat Autònoma de Barcelona, 2006.

Estudio del overhead en la migración de algoritmos paralelos de cluster y multicluster a GRID.

Ismael Rodriguez [1], Adrian Pousa [2], Jose Pettoruti [3], Franco Chichizola [4], Marcelo Naiouf [5], Laura De Giusti [6], Armando De Giusti [7]

Instituto de Investigación en Informática (III-LIDI) – Facultad de Informática – UNLP

Abstract

An experimental work is presented, in which a multicluster application has been migrated. In this application, communications among processes/processors was handled directly from MPI to the same architecture with Grid support (Globus Toolkit 4.04) in order to study the overhead generated by the incorporated middleware.
The N-queens solution with static distribution has been chosen and two clusters interconnected with Grid support, including the GridFTP, RFT, GRAM, WS-GRAM, RLS, and MDS services, were configured. Later, the response time for the N variable was studied, analyzing the speedup and efficiency.
The most interesting result is the overhead independence generated by the Grid middleware when scaling the problem.
Keywords: *Clusters. Grid. Grid Services. Parallel Algorithms. N-Queens.*

Resumen

Se presenta un trabajo experimental, en el que se migró una aplicación multicluster en la que las comunicaciones entre procesos/procesadores se manejaba directamente desde MPI a la misma arquitectura con soporte Grid (Globus Toolkit 4.0.4) con el objetivo de estudiar el overhead generado por el middleware incorporado.
Se eligió la solución de N reinas con distribución estática y se configuraron dos clusters interconectados con el soporte para Grid que incluía los servicios GridFTP, RFT, GRAM, WS-GRAM, RLS, MDS. Posteriormente se estudió el tiempo de respuesta para N variable, analizando el speedup y la eficiencia.
El resultado de mayor interés es la independencia del overhead generado por el middleware de Grid al escalar el problema.
Palabras Clave: *Clusters. Grid. Servicios Grid. Algoritmos paralelos. N-Reinas.*

VI Workshop de Procesamiento Distribuido y Paralelo.

[1] Becario Alumno Telefónica e III-LIDI Aux. Docente de la Facultad de Informática UNLP. ismael@lidi.info.unlp.edu.ar
[2] Becario Alumno Telefónica e III-LIDI Aux. Docente de la Facultad de Informática UNLP. apousa@ lidi.info.unlp.edu.ar.
[3] Becario Alumno III-LIDI. josep@ lidi.info.unlp.edu.ar.
[4] Becario de Doctorado del CONICET. Profesor Adjunto de la Facultad de Informática UNLP. francoch@lidi.info.unlp.edu.ar.
[5] Profesor Titular D.E. Facultad de Informática UNLP. mnaiouf@lidi.info.unlp.edu.ar.
[6] Profesor Adjunto de la Facultad de Informática UNLP. ldgiusti@lidi.info.unlp.edu.ar
[7] Investigador Principal CONICET. Profesor Titular de la Facultad de Informática UNLP. degiusti@lidi.info.unlp.edu.ar.

[*] *Esta investigación es parcialmente financiada porTelefónica de Argentina, CIC, CyTED y Fundación YPF.*

1. INTRODUCCIÓN

La utilización de arquitecturas como Clusters, Multiclusters y Grid, comunicadas vía mensajes y soportadas por redes de diferentes características y topologías se ha generalizado, tanto para el desarrollo de algoritmos paralelos como para el de servicios WEB distribuidos [1] [2].

Un *cluster* es una clase de sistema de procesamiento distribuido compuesto por un conjunto de máquinas *stand-alone* interconectadas trabajando cooperativamente como un recurso de cómputo único e integrado [3]. Al conectar dos o más clusters sobre una LAN o WAN se tiene alguna forma de *multicluster*.

Diferentes variantes de multicluster se obtienen si todos los clusters están sobre una misma red o enlazan diferentes redes; si el soporte de sistema operativo es común a todos los componentes; si cada cluster es homogéneo o heterogéneo; si la red de comunicaciones tiene un ancho de banda fijo entre nodos o es variable (típico de una WAN utilizando Internet) y si cada cluster está dedicado a la aplicación definida para el multicluster o la comparte con otras tareas.

Un *Grid* es un tipo de sistema paralelo y distribuido que permite compartir, seleccionar y agregar recursos autónomos distribuidos geográficamente tales como computadoras, software, datos, bases de datos, dispositivos especiales, instrumentos y personas. Esta configuración colaborativa depende de la disponibilidad, capacidad, costo y requerimientos del usuario [4]. También un Grid puede definirse como un entorno virtual de procesamiento de información donde el usuario tiene la "ilusión" de un único y potente recurso de cómputo que en realidad se encuentra distribuido [5].

Actualmente existen numerosas áreas de aplicación para estas arquitecturas, tales como: cómputo científico, simulación, modelos industriales, medicina, comercio electrónico, manejo de bases de datos distribuidas, Internet (portales, Web services), E-Government o aplicaciones críticas tales como reactores nucleares, bancos, armas militares o control industrial en tiempo real. [6].

Algunas características de un entorno Grid [7]:

- Los recursos y servicios pueden unirse o dejar el Grid dinámicamente.
- El Grid integra recursos (procesadores, instrumentos, bases de datos, etc) que son heterogéneos, geográficamente distribuidos y en general conectados por una WAN.
- Los recursos pueden ser accedidos *on-demand* por un conjunto de usuarios que configuran una comunidad virtual.
- El Grid está configurado usando protocolos e infraestructura de propósito general, no necesariamente común a todos sus nodos.
- Problemas clásicos de los sistemas distribuidos como la tolerancia a fallas, el balance de carga o la calidad de servicio aparecen con un grado mayor de complejidad en el Grid.

Si miramos las funcionalidades de la estructura de un Grid "por capas" podemos señalar las siguientes:

A. El nivel más bajo constituido por los servicios Grid ("Factory layer") para dar soporte a la utilización de los recursos locales (procesadores, datos, red).

B. La capa que da los servicios de autenticación y seguridad para permitir el intercambio de datos entre resursos remotos ("Connective Layer).

C. La capa de administración de recursos ("Resource Layer") que permite compartirlos y establecer una conexión lógica entre ellos.

D. La capa colectiva es la que coordina las interacciones entre múltiples recursos asociados a procesos distribuidos físicamente.

E. Por último la capa de aplicación es la que trata de resolver la interacción con el usuario, de modo que éste visualice en forma "transparente" la configuración virtual con la que trabajará.

Algunos autores consideran que un Grid es un "Cluster of Clusters", lo que resulta una definición algo restrictiva pero útil para la evolución de aplicaciones paralelas de Clusters a Grid. [8]

Pueden mencionarse algunas similitudes y diferencias:

- En un cluster normalmente se configura una única máquina paralela virtual que puede estar ejecutando una aplicación dedicada. Un Grid permite configurar múltiples máquinas paralelas virtuales para varios usuarios/aplicaciones simultáneas.
- Tanto clusters como Grids se basan en procesadores heterogéneos. Sin embargo en Grid esta heterogeneidad se extiende a la red de comunicaciones y al tipo de componentes en cada nodo que pueden ser procesadores, instrumentos, sensores, etc.
- El middleware necesario para Grid es más complejo que el de los clusters. Fundamentalmente, para configurar la máquina paralela virtual es necesario una etapa de identificación de recursos físicos y su ubicación. Además en el Grid es necesario monitorear la ejecución de tareas sobre múltiples máquinas virtuales con usuarios de diferente nivel y con distintos derechos de acceso a los recursos.
- Asimismo las herramientas para el desarrollo de aplicaciones requieren un mayor nivel de abstracción en Grid, por la complejidad y variedad de los múltiples usuarios que pueden utilizar la arquitectura.

Es interesante notar que una estructura de multicluster, visualizada como un *número limitado de clusters dedicados que cooperan en una única aplicación paralela*, es un punto intermedio entre clusters y Grid y requerirá algunos servicios especiales en su middleware (especialmente para autenticar derechos de usuarios que acceden a recursos remotos).

El desarrollo de algoritmos paralelos sobre arquitecturas débilmente acopladas y geográficamente distribuidas como multiclusters y Grid presenta nuevos desafíos, entre los que pueden mencionarse:

- La heterogeneidad de las comunicaciones y su costo variable según los nodos a conectar dificulta la asignación óptima de tareas a procesadores y el balance dinámico de la carga.
- Los modelos para predicción de performance son complejos y agregan la incertidumbre del ancho de banda efectivo en el caso de emplear Internet.
- La granularidad óptima a emplear depende de la relación entre potencia de cómputo local y remota. Muchas veces la configuración efectiva de los nodos remotos a utilizar no es conocida a priori.
- De mínima se requiere un soporte de middleware para la autenticación de los usuarios y de los recursos remotos a utilizar.
- El modelo cliente-servidor (paradigma muy empleado en algoritmos paralelos sobre clusters) se torna ineficiente al incrementar el número de nodos. Esto requiere la reformulación de algoritmos que ejecutan sobre clusters.
- Las herramientas de software más generalizadas en clusters (ej. MPI o PVM) tienen restricciones al tratar de emplearlas en topologías que conectan diferentes redes.

Estos elementos llevan al desarrollo de nuevos paradigmas, lenguajes y ambientes de software para el desarrollo de sistemas paralelos sobre multicluster y grid. Asimismo los recursos de middleware necesarios para la visión de "máquina paralela virtual" son más complejos y aún no se han estandarizado.

El desarrollo de algoritmos paralelos sobre clusters (y multiclusters dentro de la misma red) ha utilizado principalmente bibliotecas de comunicaciones como PVM y MPI [9] [10]. Sin embargo al tratar de ejecutar las mismas aplicaciones sobre un Grid surgen problemas con el manejo de estas bibliotecas de comunicaciones, que actualmente están buscando un nuevo standard para Grid. Esto se debe principalmente a la necesidad de resolver el acceso remoto en forma transparente para la operación de procesos y el acceso a los datos.

Básicamente el soporte de servicios del middleware de Grid debe proveer las siguientes funciones:

- Identificar los recursos disponibles y su ubicación física. Esto obliga a mantener información distribuida del estado de los recursos.

- Administrar datos distribuidos entre procesos/procesadores ubicados en diferentes nodos del Grid.

- Manejar interacciones (comunicación, sincronización) entre procesos que pueden estar geográficamente distribuidos.

- Administrar algún mecanismo de seguridad común, transparente para los usuarios finales.

- Controlar la ejecución de "sesiones" y "programas" independientemente de su origen y del equipamiento distribuido asociado.

El análisis por capas que se ha realizado previamente del software de base para administrar un Grid hace evidente que hay un overhead natural al pasar una aplicación que se ejecuta sobre un cluster local o sobre un multicluster dedicado con usuarios autorizados en todos los nodos a una estructura de Grid geográficamente distribuida, que integra diferentes redes locales con alguna forma de red WAN [11] [12].

En este trabajo se enfoca el problema de la migración de cluster/multicluster a Grid en el caso de algoritmos con distribución de carga estática y de alto porcentaje de procesamiento en relación con las comunicaciones.

Se ha elegido este tipo de problema "simple" para poder identificar claramente el overhead "básico" que genera el middleware de Grid. En particular se eligió el caso de N-Reinas porque había sido extensamente estudiado previamente por los autores sobre cluster y multicluster, lo que permitía trabajar sobre un código probado y con resultados conocidos [13] [14] [15].

Se configuró un GRID experimental, con todos los componentes del middleware conectando 2 clusters de 8 máquinas cada uno (de hecho los resultados se pueden extender fácilmente a más clusters y/o máquinas) y se estudió el overhead, sin modificar el código de base.

2. ANALISIS DEL PROBLEMA DE LAS N-REINAS

El problema de las N-reinas consiste en ubicar N reinas en un tablero de NxN de tal manera que ninguna de ellas ataque a otra [16][17]. Una reina ataca a otra si se encuentran en la misma diagonal, fila o columna.

Solución secuencial

Una solución inicial al problema de las N-reinas, mediante un algoritmo secuencial elemental, consiste en probar todas las combinaciones posibles de ubicación de las reinas en el tablero y quedarse con aquellas que son válidas, interrumpiendo la búsqueda en el momento en que esto no se cumple. Teniendo en cuenta que una combinación válida puede generar hasta 8 soluciones diferentes, las cuales son rotaciones de la misma, se puede reducir la cantidad de distribuciones que es necesario evaluar. En esto se basa el mejor algoritmo secuencial encontrado para este problema [18] [19].

El algoritmo realiza N/2 iteraciones, y en cada una de ellas ubica la reina en una posición diferente de la primera fila. Las N/2 posiciones restantes no son evaluadas debido a que son combinaciones simétricas de las anteriores.

A partir de la reina ubicada en la primera fila se determina el vector de posiciones válidas para la siguiente fila, y para cada una de ellas se determinan las soluciones que las mismas generan (Figura 1.a). Para determinar la cantidad de soluciones a partir de la fila i (tal que toda fila j con $j \leq i$ tiene ubicada su reina), se determina el vector de posiciones válidas para la fila i+1, donde para cada una de ellas se vuelve a repetir este paso. Esto continúa hasta llegar a ubicar una reina en la última fila, o cuando no hay más posiciones válidas en una cierta fila (Figura 1.b). Al llegar a ubicar una reina en la última fila se calcula la cantidad de soluciones diferentes que generan dicha combinación y su simétrica al ser rotadas 90º, 180º y 270º (Figura 1.c).

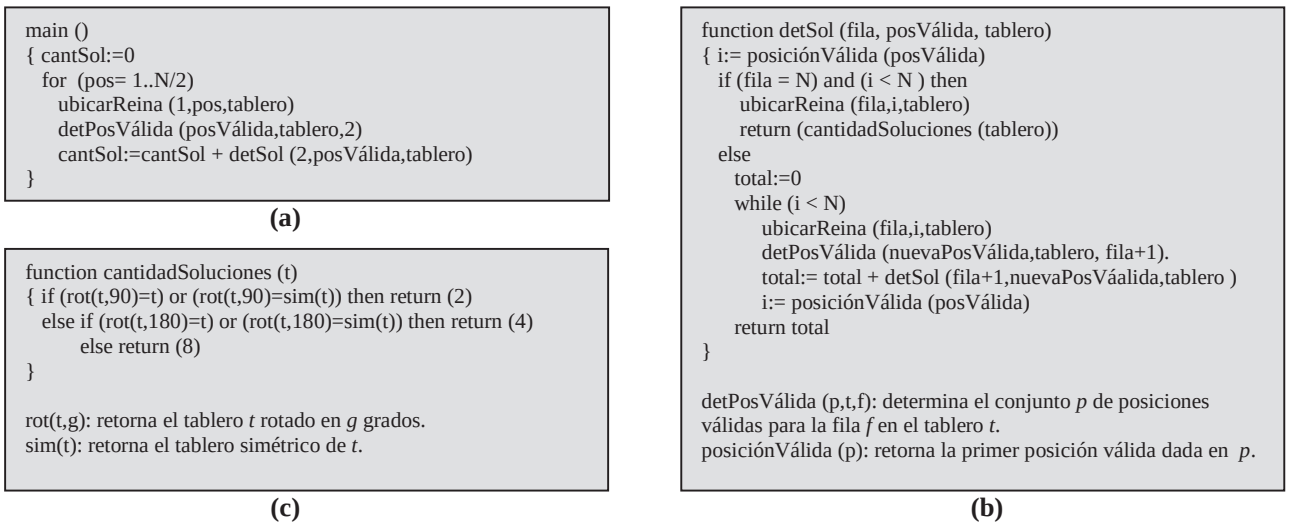


Figura 1. Pseudocódigo de la solución secuencial

Solución paralela

Para la solución paralela de este problema, se ubica la reina en una o más filas y se obtienen todas las soluciones para esa disposición inicial. Cada procesador se encarga de resolver el problema para un subconjunto de éstas, de manera tal que el sistema completo trabaje con todas las posibles combinaciones de esas filas [13] [14]. Para esto se usan las cuatro primeras filas para formar cada una de las combinaciones a resolver. De esta manera se obtienen N^4 combinaciones diferentes para distribuir entre todos los procesadores heterogéneos, siendo N el tamaño del tablero.

Se ha elegido trabajar con el paradigma Master-Slave. La Figura 2.a muestra el pseudocódigo del algoritmo utilizado por el proceso Master, el cual reparte todas las combinaciones entre los procesos Slaves al comenzar la aplicación (distribución estática), resuelve un subconjunto de ellas, y luego junta los resultados de todos los slaves. A su vez la Figura 2.b muestra el pseudocódigo de los procesos Slaves los cuales calculan las posibles soluciones al problema a partir de las combinaciones que le son asignadas, y por último la Figura 2.c muestra el pseudocódigo de un módulo utilizado para determinar las soluciones a partir de una cierta configuración inicial.

En esta clase de problemas el tiempo de comunicación entre procesos *Tc* no es significativo frente al tiempo de procesamiento local *Tp* (*Tp >> Tc*). Esta característica permite identificar el overhead generado por el middleware utilizado para GRID sin que las comunicaciones influyan en este análisis.

En este punto vale aclarar que en trabajos previos se han realizado algoritmos más eficientes al utilizar distribuciones dinámicas o semidinámicas que logran un mejor balance de carga a costa de un aumento en la cantidad de comunicaciones [15].

```
main proceso 0 ()
{ //Envia a cada procesador las combinaciones que debe evaluar
  scatter (rangos,0)
  //Procesa su parte
  cantSol:=0
  while  (procesador 0 tenga combinaciones)
    determina la ubicación en la fila 1 (p1)
    determina la ubicación en la fila 2 (p2)
    determina la ubicación en la fila 3 (p3)
    determina la ubicación en la fila 4 (p4)
    cantSol:= cantSol + detSolParcial (p1,p2,p3,p4, tablero)
  //Recibe los resultados de todas las tareas
  reduce(cantSol, ADD, total,0)
 }
```

(a)

```
main procesador i ()   // Con i > 0
{ //Recibe las combinaciones a realizar
  scatter (rangos,0)
  //Procesa su parte
  cantSol:=0
  while  (procesador i tenga combinaciones)
      determina la ubicación en la fila 1 (p1)
      determina la ubicación en la fila 2 (p2)
      determina la ubicación en la fila 3 (p3)
      determina la ubicación en la fila 4 (p4)
      cantSol:= cantSol + detSolParcial (p1,p2,p3,p4,tablero)
  //Envia los resultados de todas las tareas
  reduce(cantSol, ADD, total,0)
 }
```

(b)

```
function detSolParcial (posFila1, posFila2, posFila3, posFila4, tablero)
{ ubicarReina (1,posFila1,tablero)
  ubicarReina (2,posFila2,tablero)
  ubicarReina (3,posFila3,tablero)
  ubicarReina (4,posFila4,tablero)
  detPosVálida (posVálida, tablero,5).
  return detSol  (5, posVálida, tablero)
}

detPosVálida (p,t,f): determina el conjunto p de posiciones válidas para la
fila f del tablero t.
```

(c)

Figura 2. Pseudocódigo de la solución paralela. La función detSol es la descripta en la Figura 1.b.

3. CONFIGURACION EXPERIMENTAL UTILIZADA

Se configuró una GRID utilizando dos clusters (“Foster Cluster” y “Newton Cluster”) del Instituto, añadiéndole a cada cluster un nodo cabecera (*head node)* como nodo GRID utilizando Globus Toolkit 4.0.4 como GRID Middleware. Ademas se agregaron servidores DNS, Web (*GRID Portal)* y un Router. La Figura 3 muestra gráficamente la arquitectura configurada.

Los clusters utilizados poseen las siguientes características técnicas:

Foster Cluster	***Newton Cluster***
8 Nodos PC [Foster1..Foster8] Intel Pentium III 700 MHz 256 MB RAM HD 20 GB Fast Ethernet 100 Mbps LAM-MPI 7.1.3	*8 Nodos PC [Newton1..Newton8]* Intel Pentium 4 2,66 GHz 512 MB RAM HD 40 GB Fast Ethernet 100 Mbps LAM-MPI 7.1.3
Head Node [Foster] Intel Pentium 4 2,66 GHz 512 MB RAM HD 80 GB Fast Ethernet 100 Mbps LAM-MPI 7.1.3	*Head Node [Newton]* Intel Pentium 4 2,66 GHz 512 MB RAM HD 80 GB Fast Ethernet 100 Mbps LAM-MPI 7.1.3

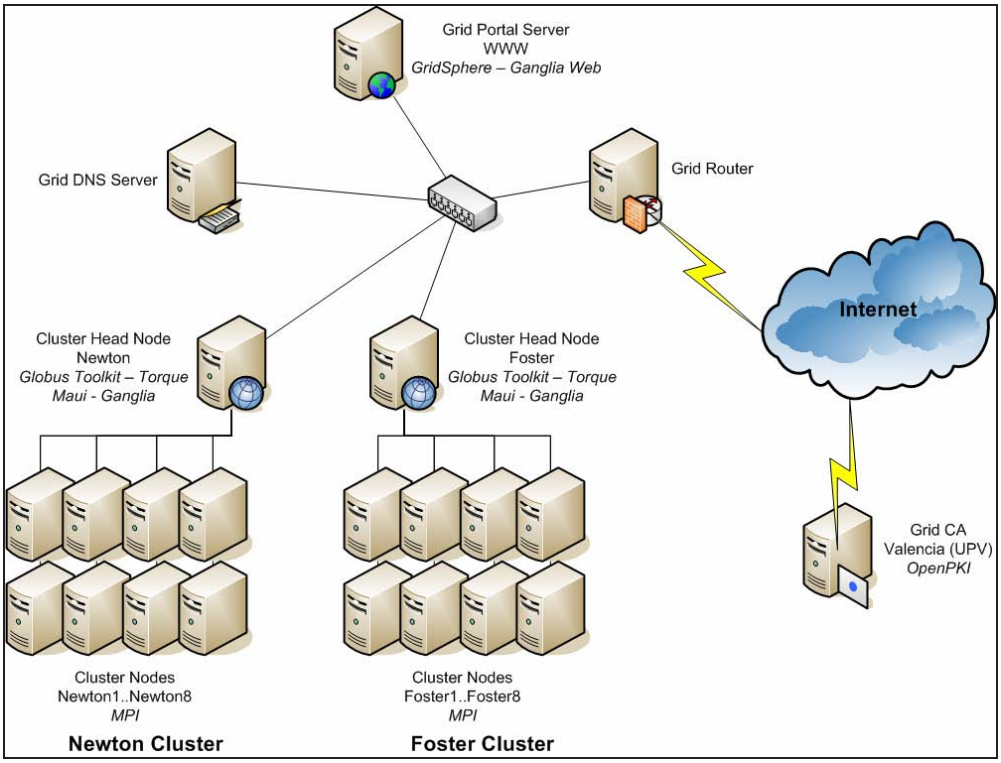


Figura 3. Infraestructura y topología de la III-LIDI GRID.

En ambos head nodes se utilizó Globus Toolkit 4.0.4 como Grid Middleware utilizando todos sus componentes (GridFTP, RFT, GRAM, WS-GRAM, RLS, MDS).

Se utilizó LAM-MPI 7.1.3 como librería de pasaje de mensajes en ambos clusters. Además, se utilizó Torque Resource Manager como manejador de recursos y Maui Cluster Scheduler. Estos servicios administran los recursos y solicitudes de procesamientos sobre los clusters.

Se utilizó la entidad certificante (CA) del proyecto CYTED, ubicada físicamente en la Universidad Politécnica de Valencia (UPV).

Para integrar la infraestructura multicluster al GRID, se configuró WS-GRAM interactuando con Torque logrando así el acceso a los recursos de los cluster a través del GRID.

Como herramienta de monitoreo del GRID se configuró Ganglia y su front-end web. Además, se instaló el Portal Web GridSphere para facilitar el uso del GRID a los usuarios.

La secuencia de ejecución sobre la infraestructura GRID es la siguiente:

- Se tiene un programa paralelo implementado en lenguaje C, empleando la biblioteca MPI.
- Se divide la ejecución del programa entre los dos *head nodes*, Foster y Newton.
- Cada nodo recibe el pedido de ejecución de su parte en paralelo a través de WS-GRAM.
- En cada *head node*, WS-GRAM interactúa con el Resource Manager “Torque” y el Cluster Scheduler “Maui” para procesar la ejecución.

- El Resource Manager de cada nodo interactúa a su vez con el ambiente MPI ejecutando el programa sobre el cluster.
- Cuando la ejecución sobre el cluster en Foster finaliza, éste envía a través de GridFTP los resultados a Newton.
- Cuando Newton finaliza la ejecución, lee los resultados enviados por Foster y corre un tercer programa que realiza la unión de los resultados de ambos clusters.

4. RESULTADOS OBTENIDOS

La Tabla I muestra los tiempos (en segundos) requeridos para la ejecución secuencial, paralela sobre la arquitectura de cluster y paralela sobre una arquitectura GRID para tableros de tamaño 17x17, 18x18, 19x19 y 20x20.

Tamaño	Tiempo Secuencial	Tiempo utilizando Cluster	Tiempo utilizando GRID
17	51,63	19,25	41,32
18	385,25	136,18	155,62
19	3025,65	975,77	997,21
20	24478,21	8547,53	8570,07

Tabla I. Tiempos secuencial y paralelo (Cluster y GRID) de las pruebas realizadas.

La Figura 4 muestra el overhead en segundos resultante del empleo del middleware de Grid para los diferentes tamaños de tablero. Se puede observar que el overhead prácticamente no depende del trabajo realizado (que crece exponencialmente con el tamaño de los tableros).

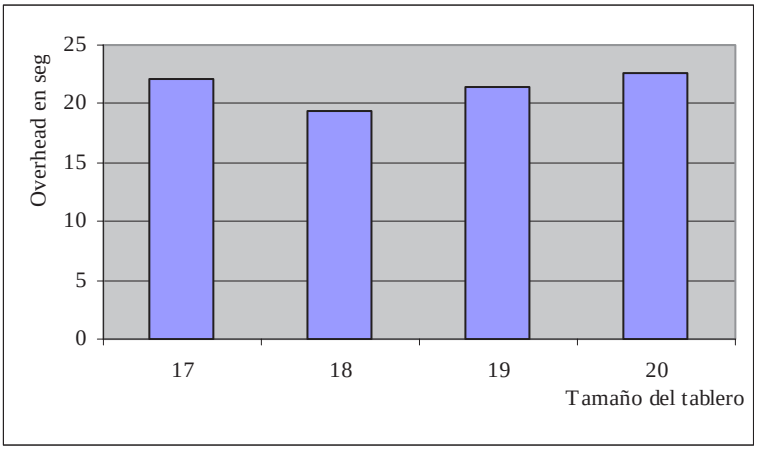


Figura 4. Overhead producido por el middleware de GRID.

Para analizar el rendimiento del algoritmo en la arquitectura paralela se utiliza la métrica speedup dada por la fórmula:

$$Speedup = \frac{TiempoSecuencial}{TiempoParalelo}$$

En el caso de una arquitectura heterogénea el "Tiempo Secuencial" está dado por el tiempo del algoritmo secuencial ejecutado en la máquina con mayor potencia de cálculo.

Para evaluar cuan bueno es el speedup obtenido se compara con el speedup teórico de la arquitectura sobre la cual se está trabajando. El mismo considera la potencia de cálculo relativa de cada máquina con respecto a la potencia de la máquina más potente.

$$SpeedupTeórico = \sum_{i=1}^{B} P_i$$

donde

B es la cantidad de máquinas que componen la arquitectura utilizada.

P_i es la potencia de cálculo relativa de la máquina *i* con respecto a la potencia de la mejor máquina. Esta relación se expresa en la fórmula a continuación:

$$P_i = \frac{tiempoSecuencial(máquinaMasPotente)}{tiempoSecuencial(m_i)}$$

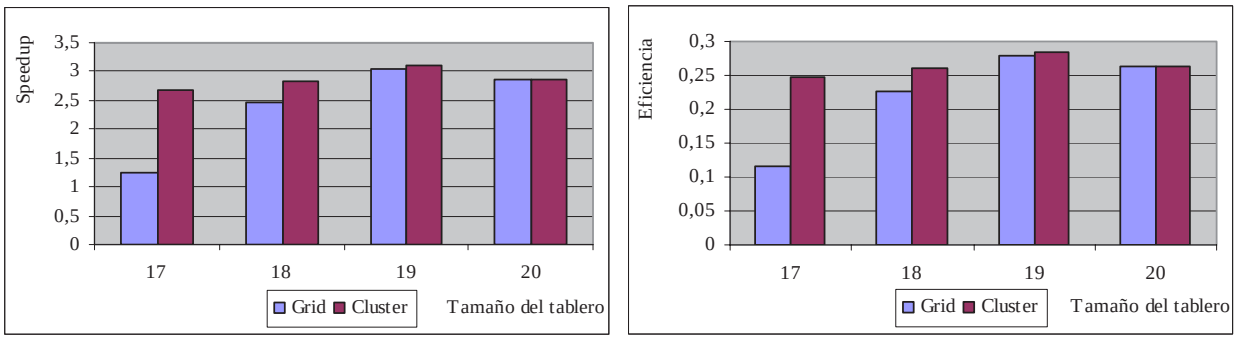
Como se dijo anteriormente la arquitectura utilizada está formada por 2 clusters de 8 máquinas cada una donde los procesadores que componen uno de los clusters poseen un 36% de la potencia de cálculo de los que pertenecen al otro cluster. Por lo tanto el speedup teórico de la arquitectura es 10,88 (0,36*8 + 1*8).

La Tabla II muestra el speedup y la eficiencia obtenidos en cada una de las arquitecturas para los diferentes tamaños de tablero.

	Grid		**Cluster**	
Tamaño	**Speedup**	**Eficiencia**	**Speedup**	**Eficiencia**
17	1,249	0,114	2,681	0,246
18	2,475	0,227	2,828	0,260
19	3,034	0,278	3,100	0,284
20	2,856	0,262	2,863	0,263

Tabla II. Speedup y eficiencia de las pruebas realizadas

Por último en las Figuras 5.a y 5.b se visualizan los valores mostrados en la Tabla II.



(a) **(b)**

Figura 5. (a) Speedup y (b) Eficiencia de las pruebas realizadas en Cluster y GRID.

5. CONCLUSIONES Y LINEAS DE TRABAJO FUTURO

El resultado más importante es que para esta clase de aplicación (de alto porcentaje de tiempo de procesamiento respecto del tiempo de comunicación) el overhead que introduce el software de Grid representa prácticamente un tiempo constante, que no cambia al escalar el problema (de N=17 a N=20).

Otro aspecto interesante es que la migración de la aplicación es prácticamente transparente, manteniendo el código original y agregando los servicios de autenticación, ejecución remota y comunicación inter-cluster a través del middleware de Grid. Los resultados obtenidos respecto del overhead básicamente no dependen del número de clusters que se conecten en el Grid, ni tampoco del número de máquinas de cada uno de ellos.

Actualmente se está trabajando en analizar problemas con mayor comunicación y con alguna forma de balanceo dinámico en la ejecución (lo que obliga a una interacción durante la ejecución entre procesos en diferentes nodos del Grid). Asimismo se estudia la eficiencia alcanzable al escalar el problema y la dimensión de los clusters.

REFERENCIAS

[1] Z. Juhasz (Editor), P. Kacsuk (Editor), D. Kranzlmuller (Editor). “Distributed and Parallel Systems: Cluster and Grid Computing”. The International Series in Engineering and Computer Science. Springer; 1 edition. 2004.

[2] Di Stefano M. “Distributed data management for Grid Computing”. John Wiley & Sons Inc. 2005.

[3] Grid Computing and Distributed Systems (GRIDS) Laboratory - Department of Computer Science and Software Engineering (University of Melbourne). “Cluster and Grid Computing”. 2007. http://www.cs.mu.oz.au/678/.

[4] Joseph J., Fellenstein C. “Grid Computing”. On Demand Series. IBM Press. 2003.

[5] Foster I., Kesselman C., Kaufmann M. “The Grid 2: Blueprint for a New Computing Infrastructure”. The Morgan Kaufmann Series in Computer Architecture and Design. 2 edition. 2003.

[6] “CSM23 Grid Computing”. http://www.computing.surrey.ac.uk/courses/csm23.

[7] Berman F., Fox G., Hey A. “Grid Computing: Making The Global Infrastructure a Reality”. John Wiley & Sons. 2003.

[8] De Giusti A. et al. “Parallel algorithms on Multi-Cluster Architectures using GRID Middleware. Experiences in Argentine Universities”. Proceedings of the I Iberian Grid Infrastructure Conference. Spain. 2007. Pag. 322-332

[9] “PVM Parallel Virtual Machine”. http://www.csm.ornl.gov/pvm/

[10] “MPI Message Passing Interface”. http://www.mpi-forum.org/

[11] “Grid Computing Info Centre”. http://www.gridcomputing.com/

[12] “The Globus Alliance”. http://www.globus.org/

[13] De Giusti L., Novarini P., Naiouf M. R., De Giusti A. E. “Parallelization of the N-queens problem. Load unbalance analysis”. Workshop de Procesamiento Paralelo y Distribuido. 2003. pag.397-403

[14] De Giusti Laura, Naiouf Marcelo, Chichizola Franco, De Giusti Armando. Informe Técnico III-LIDI “Balance de Carga en procesamiento Multicluster. Aplicación al problema de las N-Reinas”. 2005.

[15] Naiouf M. R., De Giusti L. C., Chichizola F., De Giusti A. E. “Dynamic Load Balancing on Non-homogeneous Clusters”. G.Min et al. (Eds.): ISPA 2006 Ws, LNCS 4331, pags. 65-73. Springer – Verlag. Berlin Heidelber 2006.

[16] Bruen A, Dixon R. “Then n-queens problem. Discrete mathematics”. 12:393-395. 1997.

[17] Hedetniemi S, Hedetniemi T, Reynolds R. “Combinatorial problems on chessboards: II”. Chapter 6 in domination in graphs: advanced topic, pág 133-162. 1998.

[18] Somers J. “The N-queens problem a study in optimization”. www.jsomers.com/nqueen_demo /nqueens.html.

[19] Takaken. “N-queens problem (number of solutions)”. http://www.ic-net.or.jp/home/takaken/e/queen/.

Exclusión Mutua para Grupos de Procesos utilizando un actor

Karina M. Cenci * Jorge R. Ardenghi **

Laboratorio de Investigación en Sistemas Distribuidos
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Resumen

Las aplicaciones distribuidas están formadas por un conjunto de procesos, los cuales pueden competir por utilizar un recurso o trabajar en forma conjunta para resolver una tarea.

Estas aplicaciones requieren protocolos que permitan concurrencia entre los procesos que trabajan cooperativamente y exclusión mutua para aquellos que compiten por utilizar el recurso. En este trabajo se presenta un algoritmo para exclusión mutua para grupos de procesos basándose en el modelo de memoria compartida asincrónica, donde cada conjunto de procesos que realice una actividad conjunta compiten conjuntamente para acceder al recurso.
...

Palabras Claves: Sistemas Distribuidos - Exclusión Mutua - Exclusión Mutua para Grupos de Procesos - Concurrencia -

*e-mail: kmc@cs.uns.edu.ar
**e-mail: jra@cs.uns.edu.ar

1. Introducción

Las aplicaciones utilizan recursos, algunas requieren uso exclusivo de los mismo y otras que realizan trabajo cooperativo comparten el acceso a esos recursos. A partir de estos requerimientos, es necesario considerar en las aplicaciones distribuidas protocolos que soporten las características de *exclusión mutua* y *concurrencia*.

- Exclusión Mutua: garantiza el acceso exclusivo a un recurso común sobre un conjunto de procesos compitiendo.
- Concurrencia: permite que los procesos no conflictivos compartan un recurso para incrementar la performance del sistema.

La característica de exclusión mutua es el primer problema que surge en los sistemas multiprogramados, en la literatura hay muchos protocolos propuestos que garantizan esta propiedad, como por ejemplo en [2], [3], [4], [7], [10], ..., algunos están basado en el modelo de memoria compartida distribuida y otros en pasaje de mensajes.

Para considerar la situación que varios procesos comparten el acceso a un recurso, se extiende el problema original de *exclusión mutua* a *exclusión mutua para grupos de procesos*. En este trabajo se consideran protocolos para la extensión a grupos de procesos basándose en el modelo de memoria compartida, dónde los procesos se comunican mediante la escritura y lectura de variables compartidas.

Se considera un conjunto de n procesos $p_0, p_1, .., p_{n-1}$ los cuales trabajan en forma independiente o en forma cooperativa en una actividad que utiliza un recurso compartido. En el resto del trabajo se considera que la actividad compartida se realiza en un grupo con el mismo interés.

Los procesos pueden participar de cualquiera de los diferentes m grupos G_0, G_1, ... , G_{m-1}. Cada uno de los grupos, cuando se encuentra activo utiliza el/los recurso/s por los cuales compiten en el sistema. En un determinado instante, sólo un grupo puede estar utilizando el recurso compartido. En el grupo, participan todos los procesos que están interesados en la actividad.

Inicialmente cada uno de los procesos está trabajando individualmente. Cuando desea trabajar en equipo, elige el *grupo*. Se considera que cada proceso trabaja en equipo por un tiempo finito, y que puede participar en cualquiera de los diferentes grupos. En la figura 1, se observan 2 formas diferentes de modelar el mismo problema. En el caso (A), los procesos P_1, P_2 y P_7 que están actualmente vinculados al grupo G_3; y los procesos P_0 y P_8 están integrando el grupo G_1. La competencia para acceder al recurso la ganó el grupo G_3, el mismo se encuentra en la sección crítica y todos los procesos vinculados están trajando cooperativamente, el grupo G_1 está compitiendo por alcanzar el permiso de utilizar el recurso. En el caso (B), los procesos P_1, P_2 y P_7 eligieron al grupo G_3 para trabajar concurrentemente, y los procesos P_0 y P_8 eligieron al grupo G_1 para trabajar concurrentemente. Cada proceso compitió por acceder al recurso, uno de los procesos que eligió al grupo G_3 obtiene el permiso entonces permite que accedan los otros procesos que quieren realizar la misma actividad cooperativamente.

En el caso, en que se considerara que cada grupo estuviera formado por un solo proceso, entonces el problema se reduciría al modelo convencional de exclusión mutua para n procesos, donde solamente un proceso a la vez puede estar en la sección crítica. Para resolver este problema se requiere una extensión del problema de la exclusión mutua al caso donde k procesos pueden

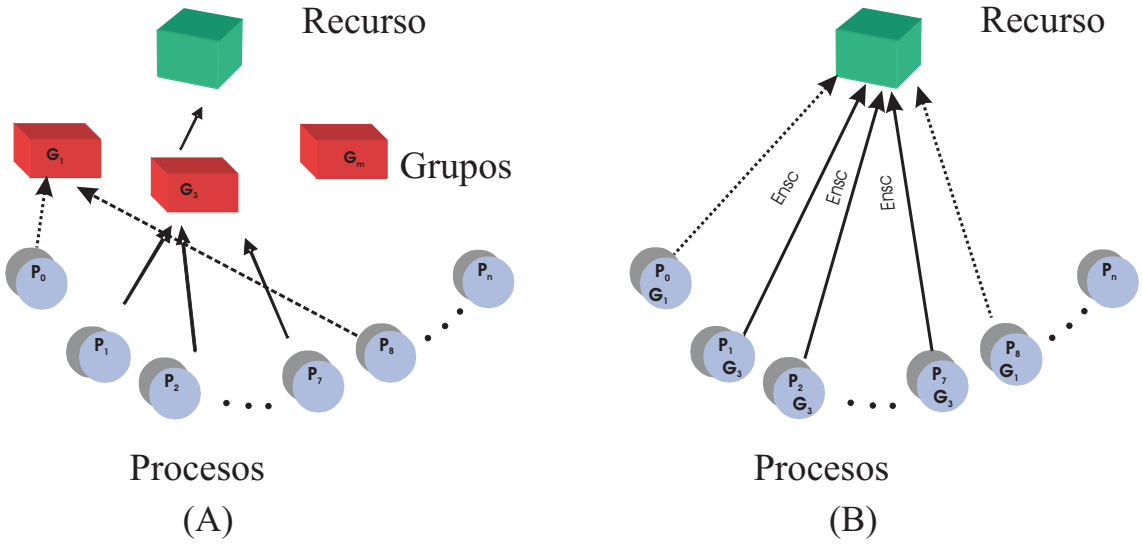


Figura 1: Ejemplo de Concurrencia y Competición

compartir la utilización del recurso en un instante de tiempo. Se requiere un algoritmo que satisfaga los siguientes requerimientos:

- *Exclusión Mutua*: si algún proceso está trabajando en un grupo, no puede haber otro proceso trabajando en un grupo diferente simultánea-mente.
- *Demora Limitada (libre de inanición)*: un proceso que desea participar de un grupo eventualmente tendrá éxito.
- *Entrada Concurrente*: si algunos procesos están interesados en un grupo y no hay un proceso interesado en otro grupo, entonces los procesos pueden participar concurrentemente del grupo.
- *Libre de interbloqueo*: cuando la sección crítica está disponible, los grupos no deberían esperar indefinidamente y alguno debería obtener el permiso para acceder.

2. Algoritmo con dos actores

El algoritmo presentado en [5] resuelve el problema de exclusión mutua para grupos de procesos utilizando dos tipos de actores: los *procesos* y los *grupos*, que se integran para competir por la utilización de un recurso. En el modelo, se tienen dos componentes que formarán parte del algoritmo, el proceso que selecciona un grupo de trabajo, y el grupo que compite para acceder a la sección crítica.

Cuando un actor no está involucrado de ninguna manera con el recurso, se dice que está en la región *resto*. Para obtener la admisión a la región crítica, un actor ejecuta un protocolo de entrada (*trying*), después que utiliza el recurso, se ejecuta un protocolo de salida (*exit*). Este procedimiento puede repetirse, de modo que cada actor sigue un ciclo, desplazándose desde la *región resto (R)*,

```
Proceso_i
... Sección Resto
Entrada_i
    Selección del grupo en g
    Si inactivo(g) entonces
      lista(g,i) = <2, espera> {Es el primer proceso en el grupo, habilita mientras está en la sección crítica que otro procesos puedan
                                participar concurrentemente en la misma}
    sino
      lista(g,i) = <1, espera> {Por lo menos hay otro proceso que estaba en el grupo}
    fin si
    Waitfor (flag(g) = etapas + 1) ∧ ((lista(g,i)=<2, espera>) ∨ (∃ j: 1..n, lista(g,i)=<2, en_cs>))
    lista(g,i) = < .., en_cs>
... Sección Crítica
Salida_i
    lista(g,i) = <0, resto>

inactivo(g) ≡ (flag(g) = 0) {Indica que el grupo está en la región resto}
```

Figura 2: Actor Proceso

```
Grupo_i
Entrada_i
    waitfor [∃ j: 1..n, lista[k,j] = < .., espera>]
    Bucar_lider(lista,i)
    para k = 1 hasta etapas hacer
      flag(i) = k {representa los diferentes niveles}
      Si (role(i,k)≠0) ó (i≤m-k) entonces
            Waitfor [∀ j ∈ oponentes(i,k) : flag(j) < k] ó [turn(comp(i,k))≠role(i,k)]
    flag(i)= etapas + 1 {para que el proceso sepa que está en la S.C.}
... Sección Crítica

Salida_i
    Waitfor [∀ j: 1..n, lista(i,j)≠<..,en_cs>]

    flag(i) = 0
```

Figura 3: Actor Grupo

a la *región de entrada (T)*, luego a la *región crítica (C)* y por último a la *región de salida (E)*, y luego vuelve a comenzar el ciclo en la *región resto*.

Como se observa en la figura 2, el primer paso que realiza el *actor proceso*, en la región de entrada, es seleccionar el grupo en el cual desea participar del conjunto de m grupos. El segundo paso es esperar hasta que el grupo seleccionado entre en la región crítica para que pueda acceder a la misma. Cuando finaliza su actividad, sale de la región crítica, se desvincula del grupo (región de salida).

Como se observa en la figura 3, el *actor grupo* inicialmente está inactivo, en la región resto, esto representa que ningún proceso lo ha seleccionado para participar en el mismo. El primer proceso que lo selecciona para participar, hace que comience la competencia por entrar en la región crítica, y se lo identifica como el primer proceso que pertenece al grupo; pasa a la región de entrada. Todos los procesos que lo seleccionen mientras se encuentra en competición por entrar a la región crítica, se agregan a los procesos ya existentes. En el caso que el grupo esté en la región crítica, si el proceso que activó al grupo está trabajando en la misma, entonces el proceso se incorpora, sino se pone en cola de espera hasta que termine la actual vuelta (todos los procesos que están trabajando finalicen su tarea y el grupo salga de la región crítica), se reinicie el ciclo, esto es, compita nuevamente por el ingreso en la región crítica.

Los *actores grupos* compiten por alcanzar el permiso de utilizar el recurso (acceso a la región

∀ flag(i): $0 \leq i \leq (m-1)$, flag(i) = 0 inicialmente
para cada cadena binaria x de a lo sumo longitud etapas-1

turn(x) inicialmente arbitraria, escrito y leído por exactamente aquellos grupos i para los cuales x es un prefijo de la representación binaria de i.

∀ lista(i,j): $0 \leq i \leq (m-1)$, $0 \leq j \leq (n-1)$,
lista(i,j) = <0, resto> inicialmente

etapas = (Si truncar(log(m)) = log(m) entonces = truncar(log(m)) sino = truncar(log(m)) + 1 fin si)

Figura 4: Algoritmo dos actores - variables compartidas

Notas

• comp(k,l)→ el nivel l del grupo k, es la cadena que consiste de los (etapas-l) bits de mayor orden de la representación binaria de k.

• etapas → está representando la cantidad de bits necesaria para almacenar hasta el valor (m-1)

• role(k,l) → el rol del $grupo_k$ en el nivel l de competeción del mismo, es el bit (etapas-l+1) de la representación binaria de k (representa si desciende de la rama derecha o izquierda)

• oponentes(k,l) → los oponentes del $grupo_k$ en el nivel l de competición, es el conjunto de índices de grupos con el mismo orden de bits en (etapas-l) y el opuesto (etapas-l+1)

crítica), sólo un único grupo tiene derecho de utilizar el recurso en un determinado instante de tiempo. El esquema base para la competencia de los grupos, está basado en el algoritmo de Tournament, con la extensión a m elementos, donde m no es necesariamente una potencia de 2. Las variables utilizadas se muestran en la figura 4.

Cada grupo está ocupado en una serie de competencias de O(log n) para obtener el recurso. Puede considerarse que la competición está dispuesta en un árbol de competencia binario. Las hojas corresponden a los m grupos. En las Notas se presentan las funciones que se utilizan en el algoritmo.

La idea es que el algoritmo cumpla las siguientes condiciones:

- Un único grupo está activo utilizando el recurso compartido (exclusión mutua)
- Si el recurso está disponible y un grupo quiere utilizarlo (está en espera) que acceda al mismo sin tener más demora.
- Si un grupo está activo y el primer proceso también, todo proceso que quiera trabajar en el mismo que lo pueda realizar, de esa manera se logra un mayor nivel de concurrencia.

El algoritmo cumple con las condiciones de *buena formación, exclusión mutua y progreso* que requiere para resolver el problema de la exclusión mutua (porque está basado en el algoritmo

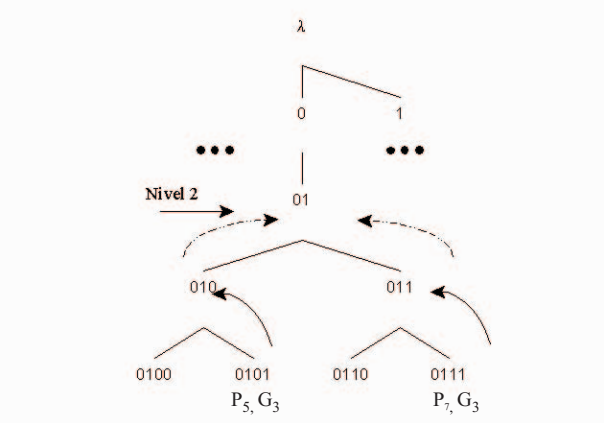


Figura 5: Procesos Concurrentes

presentado en [4]); y además satisface los requisitos de un buen algoritmo de exclusión mutua, esto es, *Libre de Interbloqueo*, *Libre de inanición* e *Imparcialidad*.

3. Presentación del Modelo

El modelo presentado está formado por un actor *proceso* que utiliza el recurso en forma compartida con otros actores *procesos* que realizan su trabajo en forma conjunta, esto es participan del mismo grupo de interés.

El actor *proceso* es el que compite por acceder al recurso participando de un determinado *grupo*. El actor *proceso* puede participar de diferentes grupos durante en el transcurso de su trabajo, pero en un determinado instante de tiempo participa de un único grupo.

En el caso (B) de la figura 1, se puede observar el comportamiento del modelo presentado. El esquema de competencia del actor *proceso* está basado en el algoritmo de Tournament, con la extensión a n elementos, donde n no es necesariamente una potencia de 2.

El proceso inicialmente está en la sección resto, cuando ingresa en la sección de entrada selecciona el grupo en el cual va a participar y comienza la competencia por acceder al recurso. La competencia se realiza por niveles, cuando supera el último nivel puede acceder a la sección crítica, si en un determinado nivel se encuentra con un proceso que quiere acceder por el mismo grupo entonces compiten conjuntamente para acceder a la sección crítica. En la figura 5, se observa la situación en la cual los procesos P_5 y P_7 que compiten en el nivel 2 comparten el mismo grupo G_3, entonces avanzan juntos.

En el modelo presentado cuando un proceso P_i compite con el proceso P_j en el nivel k identifica que es un compañero entonces espera hasta que el proceso P_j acceda al recurso y luego participar en forma cooperativa.

4. Algoritmo con un actor

El algoritmo presentado se basa en el paradigma de memoria compartida distribuida sobre las variables de control utilizadas. En la figura 6 se muestran las variables compartidas.

∀ flag(i): 0 ≤ i ≤ (n-1), flag(i) = 0 inicialmente
∀ grupo(i): 0 ≤ i ≤ (n-1), grupo(i) = -1 inicialmente
∀ oponente_compa(i,i): 0 ≤ i ≤ (n-1), oponente_compa(i) = 0 inicialmente
para cada cadena binaria x de a lo sumo longitud etapas-1

turn(x) inicialmente arbitraria, escrito y leído por exactamente aquellos grupos i para los cuales x es un prefijo de la representación binaria de i.

etapas = (Si truncar(log(n)) = log(n) entonces = truncar(log(n)) sino = truncar(log(n)) + 1 fin si)
pganador = -1

gganador = -1

Figura 6: Variables Compartidas

Las variables *flag(i)* y *grupo(i)* son escritas por el proceso P_i y leídas por el resto de los procesos; la primer variable indica el nivel de competencia del proceso y la segunda en cuál grupo está vinculada. Las variables *pganador* y *gganador* pueden ser leídas y escritas por todos los procesos; la variable *pganador* contiene la información de cuál es el primer proceso que ingresó en la sección crítica de todos los que trabajan cooperativamente y *gganandor* contiene la información de cuál es el grupo al que pertenecen los procesos que están en la sección crítica. *Etapas* contiene el número de niveles de competencia. Las variables $oponente\_compa(0..n-1, i)$ son escritas por el proceso P_i y leídas por el proceso P_j.

En la figura 7, se muestra el comportamiento del protocolo de entrada y salida en formato tradicional. Se considera los accesos a las variables compartidas atómicos. Un buen algoritmo de exclusión mutua garantiza las condiciones de buena formación, exclusión mutua y progreso. El modelo está basado en [4] que garantiza las condiciones de un buen algoritmo, se le incorporó el avance de procesos oponentes que son compañeros para acceder conjuntamente a sección crítica. En el caso que todos los oponentes sean competidores el algoritmo garantiza todas las condiciones. ¿Qué sucede si un proceso P_i en el nivel k obtiene que un oponente P_j es compañero? El estado del proceso P_i sería el siguiente:

- $flag(i) = \text{k}$
- $oponente\_compa(j, i) = 1$
- $grupo(i) = \text{g}$

El proceso P_j podría estar en las siguientes situaciones:

Nivel k	Nivel Superior	En Sección Crítica
$flag(j) = \text{k}$	$flag(j) > \text{k}$ y $<$ etapas + 1	$flag(j) =$ etapas + 1
$grupo(j) = \text{g}$	$grupo(j) = \text{g}$	$grupo(j) = \text{g}$

Si está en el mismo nivel entonces avanza al próximo nivel de competición. Si está en un nivel superior o en la sección crítica lo que modifica es cuando se encuentre en la sección de salida, ya que debe esperar a que el proceso P_i tome una decisión, acceda a la sección crítica o haya perdido el permiso de acceder directamente y tenga que continuar compitiendo.

Proceso$_i$

... *Sección Resto*

Entrada$_i$
g = seleccionar grupo
grupo(i) = g
k = 1
mientras (k $\leq$ etapas) hacer {
flag(i) = k
turn(comp(i,k)) = role(i,k)
si (role(i,k) $\neq$ 0) ó (i $\leq$ n-k) ó (i $< 2^{etapas}$ / 2) entonces
waitfor [$\forall j \in$ Oponentes(i,k) : flag(j) $<$ k] ó [Hay_Compa(i,k)] ó [turn(comp(i,k) $\neq$ role(i,k)]
Si Hay_Compa(i,k) entonces
oponente_compa(j,i) = 1
waitfor [ganar(i)] ó [$\nexists$ Oponente(i,k) : (flag(j) $\geq$ k) $\wedge$ (grupo(j) == grupo(i))]
Si ganar(i) entonces k = etapas, flag(i) = etapas + 1
oponente_compa(j,i) = 0
k = k + 1}
flag(i) = etapas + 1
Si (pganador == -1) entonces
pganador = i
gganador = grupo(i)

... *Sección Crítica*

Salida$_i$
Si (pganador == i) entonces
pganador = -1
gganador = -1
waitfor ($\forall$ j, 0 $\leq$ j $\leq$ (n-1) oponente_compa(i,j) == 0)
grupo(i) = -1
flag(i) = 0

Hay_Compa(i,k) $\equiv$ *Si*($\exists j \in$ Oponentes(i,k) : flag(j) $\geq$ k $\wedge$ grupo(j) == grupo(i)) Entonces Verdadero Sino Falso

ganar(i) $\equiv$ Si (gganador == grupo(i) $\wedge$ pganador $\neq$ -1) Entonces Verdadero Sino Falso

Figura 7: Algoritmo de un actor

Para garantizar la equidad, una vez que el primer proceso que ingresó a la sección crítica en un grupo G_l, finaliza su sección crítica no se permite que ingresen nuevos procesos vinculados al grupo G_l.

5. Medidas

Las cuestiones a tener en cuenta para obtener las medidas en la complejidad de tiempo son:

- Complejidad en un paso remoto (referencias de memoria remota) de un algoritmo es el número máximo de operaciones de memoria compartida requeridas por un proceso para ingresar y salir de su sección crítica, asumiendo que cada sentencia await es contabilizada como un única operación.
- El tiempo de respuesta del sistema es el intervalo de tiempo entre entradas a la sección crítica.

Otro factor importante para determinar la velocidad de un algoritmo es la cantidad de tráfico de interconexión que el genera. En función de este otro parámetro se define a la complejidad de tiempo de un algoritmo de exclusión mutua a ser el peor caso en el número de referencias de memoria remota por un proceso en orden para ingresar y salir de su sección crítica.

En el algoritmo presentado, cada proceso P_i compite en diferentes niveles, la cantidad máxima de niveles es del O(log(n)). En el caso que los n procesos quieran acceder a la sección crítica y no compartan trabajo, un proceso debe esperar como máximo $(n-1)$ entradas diferentes en la sección crítica.

Para poder estimar la cantidad de referencias a memoria remota, se considera que cada proceso P_i accede a las variables *flag(i)*, *grupo(i)* y *oponente_compa(i, 1 ..n)* en forma local y el resto de los accesos en forma remota (NUMA). En la siguiente tabla se muestra una comparación entre los algoritmos con dos actores y con un actor, considerando en el *algoritmo con dos actores* la cantidad de accesos del actor proceso.

Casos	Alg. dos actores	Alg. un actor
P_i quiere acceder cuando hay un compañero y es el primero	4 accesos para ingresar 1 acceso para salir	9 accesos para ingresar 1 acceso para salir
P_i es el primer proceso y no tiene oponentes	no se puede determinar	3 + log(n) + (n-1) accesos para ingresar 3 accesos para salir

En las figuras 8 y 9, se muestran algunos gráficos relacionando la cantidad de procesos, la cantidad de accesos y la proporción que hay entre ellos para el algoritmo presentado de un actor.

En otros casos, no se puede estimar la cantidad de accesos, ya que se tienen esperas ocupadas sobre variables compartidas. Para poder estimar el peor caso, se debería adaptar el algoritmo para que todas las esperas ocupadas sean locales. La adaptación introduce mayor complejidad en el algoritmo e incluye nuevas variables.

6. Conclusión

En arquitecturas distribuidas, existen aplicaciones que conviven en el ambiente que no comparten recursos ni tareas en común, pero también existen aplicaciones que colaboran para resolver un problema, o se dividen en procesos para distribuir el trabajo entre los distintos nodos y obtener un mejor rendimiento (como por ejemplo, en cálculo numérico para resolver problemas basados en matrices).

El algoritmo presentado se basa en el modelo presentado en el caso (B) de la figura 1, utilizando memoria compartida. Está compuesto por un sólo actor *proceso*. Cada proceso selecciona el grupo de interés y comienza la competición para acceder al recurso. El protocolo permite que varios procesos trabajen concurrentemente con el recurso si seleccioan el mismo grupo de interés. Se compara el algoritmo presentado *con un actor* con el algoritmo *con dos actores*. El algoritmo con dos actores presenta un mejor performance en el caso óptimo que un proceso seleccione un grupo que está en la sección crítica y el primer proceso está en la misma, en los otros casos no se puede realizar una estimación ya que realiza espera ocupada sobre variables compartidas. En cambio en el algoritmo presentado en este artículo se puede obtener una cota para el primer proceso que ingresa en la sección crítica y no tiene oponentes, esta es *3 + log(n) + (n-1)* accesos a memoria y a mayor cantidad de procesos se tiende a necesitar n accesos a memoria.

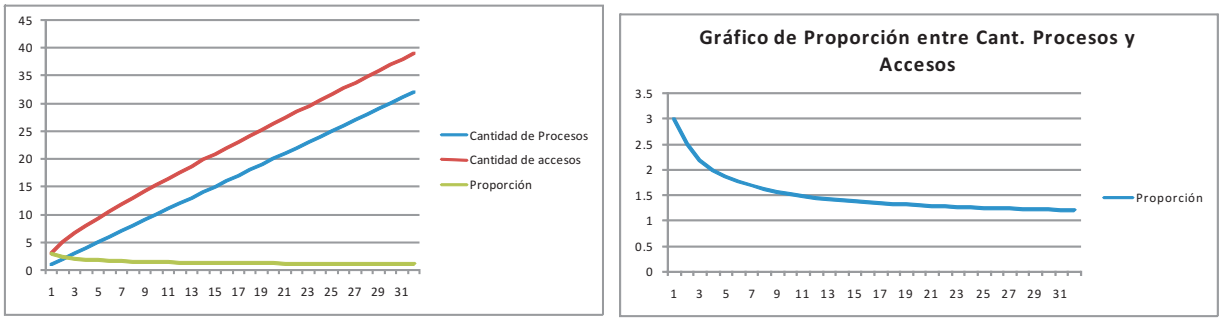


Figura 8: Con 32 procesos

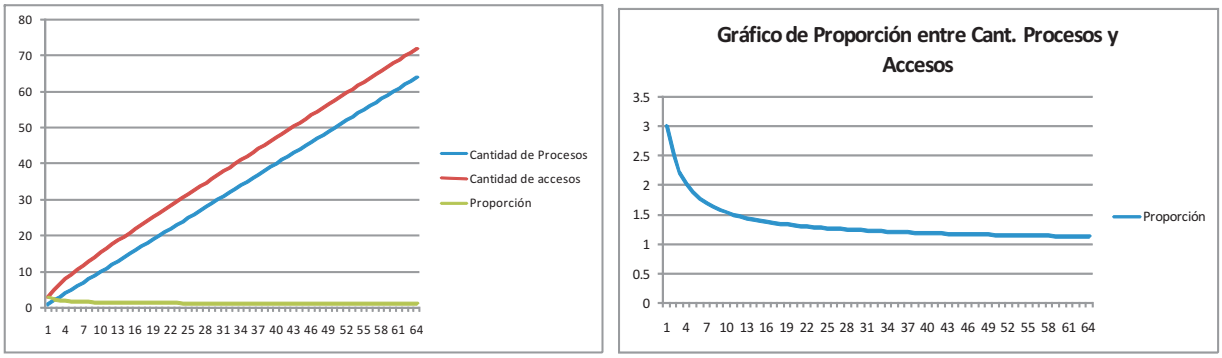


Figura 9: Con 64 procesos

Referencias

[1] M. Ben Ari. *Principles of Concurrent Programming*. Prentice Hall, Englewood Cliffs, 1982.

[2] D. Barbara, H. García-Molina. Mutual exclusion in partitioned distributed systems. *Distributed Computing*, vol. 1, no. 2, pp. 119–132, 1986.

[3] J. E. Burns, P. Jackson, N. A. Lynch, M. J. Fischer, G. L. Peterson. Data Requirements for Implementation of N-Process Mutual Exclusion Using a Single Shared Variable *Journal of the ACM*, vol. 29, issue 1, 1982.

[4] Karina Cenci, Jorge Ardenghi *Exclusión Mutua para Coordinación de Sistemas Distribuidos*. CACIC 2001.

[5] K. Cenci, J. Ardenghi *Algoritmo para Coordinar Exclusión Mutua y Concurrencia de Grupos de Procesos* CACIC 2002.

[6] Yuh-Jzer Joung *Asynchronous Group Mutual Exclusion (extended abstract)*. In Proc. 17 th. ACM PODC.

[7] L. Lamport. Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM*, Julio 1978.

[8] L. Lamport. A Fast Mutual Exclusion Algorithm. *ACM on Transactions on Computer Systems*, vol. 5, no. 1, Febrero 1987.

[9] Nancy A Lynch. *Distributed Algorithms*, 1997.

[10] M. Maekawa. A $\sqrt{N}$ Algorithm for Mutual Exclusion in Decentralized Systems. *ACM Transactions on Computer Systems*, vol 3, issue 2, pp. 145–159, Mayo 1985.

[11] Sape Mullender. *Distributed Systems*, 2da. Ed. 1993.

[12] Michael Raynal. *Algorithms for Mutual Exclusion*. MIT Press, Cambridge, 1986.

[13] Gary L. Peterson, Myths about the mutual exclusion problem. *Information Processing Letters*, Junio 1981.

[14] Silberschatz, A., y Galvin, P. *Operating System Concepts*, 5ta. ed. Addison-Wesley, 1998.

[15] Jie Wu, *Distributed System Design*, 1999.

Implementación de un Servicio Grid para el Monitoreo de Recursos Computacionales

Martín Chuburu Javier Echaiz Jorge Ardenghi

Laboratorio de Investigación de Sistemas Distribuidos (LISiDi)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Bahía Blanca (8000), Argentina
{mic,je,jrap}@cs.uns.edu.ar

Resumen

La computación grid permite administrar los recursos que se encuentran diseminados en una topología distribuida formada por nodos interconectados mediante redes de área local e Internet, con el fin de asistir a procesos que no disponen de los recursos necesarios para llevar a cabo su tarea en las máquinas locales donde fueron invocados.

Para poder hacer un uso eficiente de estos recursos es necesario tener información sobre el nivel de utilización de los mismos en cada una de las máquinas que conforman el grid, con el objetivo de poder tomar decisiones sobre la migración de los procesos que se están ejecutando en el mismo.

Contar con este tipo de información también permite la búsqueda de comportamientos anómalos como por ejemplo carga excesiva o falla en los servicios críticos. Para ello, la utilización de Servicios Grid constituye una posible herramienta para poder obtener este tipo de información y elaborar, a partir de estos servicios, una jerarquía dentro de la infraestructura grid a medida que ésta vaya creciendo.

Palabras Clave: Monitoreo de Recursos, Globus Toolkit, Servicios Grid.

1. Introducción

Un *Servicio Grid* se define como una interfaz asociada a un recurso Grid. Entonces, en un entorno Grid, un recurso y el estado asociado a éste serán administrados a través del Servicio Grid [1].

Los recursos grid pueden requerir interactuar unos con otros, además es muy probable que estos recursos se encuentren en un entorno tecnológicamente heterogéneo. Por eso es necesario un marco de trabajo que permita abstraer el servicio de mensajes entre Servicios Grid y separarlo de los detalles de implementación del entorno específico. Una *Arquitectura Orientada a Servicios* (SOA) provee tal marco de trabajo.

El *Global Grid Forum* (GGF) ha adoptado *Open Grid Service Architecture* (OGSA) basada en los principios SOA para el modelamiento de los recursos del grid, a través de Servicios Grid. Dichos servicios están construidos en base a la tecnología de Servicios Web.

La diferencia con estos últimos radica en la permanencia de los datos. En un Servicio Web no se mantiene un registro de estado entre llamadas subsecuentes al servicio. Mientras

que con Servicios Grid, a menudo es importante mantener esta información. Por lo tanto, fue necesario implementar un mecanismo complementario de forma de poder utilizar la tecnología de Servicios Web y a la vez mantener información entre un cliente y el servicio, entre una llamada y la siguiente.

Actualmente existen dos estándares disponibles para implementar Servicios Grid que cumplen con los requerimientos OGSA.

- Open Grid Services Interface (OGSI).
- Web Service Resource Framework (WSRF).

Globus Toolkit, a partir de su versión 4 (GT4), comienza a utilizar WSRF debido a que es una solución que, además de cumplir con los requerimientos de los Servicios Grid, se mantiene fiel a los fundamentos de Servicios Web.

El punto principal de discusión es la divergencia entre las especificaciones OGSI y las prácticas populares en la comunidad de Servicios Web. Algunos de los cuestionamientos hechos por esta comunidad están relacionados con la sintaxis XML utilizada por OGSI que causa conflictos con APIs estándares como JAX-RPC, con la extremada orientación a objetos en el modelo de recursos con estado que atenta contra el modelo de Servicio Web (que es "sin estado"), y con la falta de soporte para algunas versiones de WSDL. El principal objetivo en la reestructuración hacia WSRF es mantener las comunidades de Servicios Grid y Servicios Web unidas.

A diferencia de OGSI, WSRF hace una distinción explícita entre "servicio" y "recursos con estado" actuando bajo ese servicio. WSRF define los medios por los cuales un Servicio Web y un recurso con estado se componen. WSRF denomina a esta composición *WS-Resource*.

De acuerdo con WSRF, un recurso con estado tiene los datos que representan al mismo descriptos en un documento XML que es conocido y accedido por uno o más Servicios Web.

Es importante notar que para los clientes, el servicio y los recursos son vistos como una misma cosa a través del archivo WSDL que constituye la interfaz del mismo. Dichos clientes nunca tratarán directamente con instancias de los recursos sino que lo harán implícitamente a través de las interacciones con el servicio que cumple con la especificación WSRF.

La implementación de WSRF implícitamente pasa la información de identificación del recurso cuando ocurre una interacción de mensajes entre un cliente y un *WS-Resource*. El cliente no tiene que incluir explícitamente un identificador de recursos en la solicitud. En su lugar, el identificador requerido está implícitamente asociado a un intercambio de mensajes.

WS-Addressing estandariza la forma de representación de las direcciones de los Servicios Web. Tal representación es conocida como Endpoint Reference (EPR). Un EPR puede contener, además de la dirección de referencia al Servicio Web, otros metadatos asociados con el Servicio Web. Para el caso de un EPR que referencia a un *WS-Resource*, éste puede incluir un elemento que define el recurso con estado a ser utilizado en la ejecución de todos los intercambios de mensajes realizados utilizando este EPR. Este tipo de EPR se denomina *WS-Resource-qualified endpoint reference*.

En la sección 2 se listarán los pasos involucrados en la construcción de un Servicio Grid que sigue el estándar WSRF. A continuación, en la sección 3, se detallará el desarrollo de un modelo basado en Servicios Grid, que fue implementado para monitorear los recursos computacionales de un cluster de máquinas. En la sección 4 se mencionarán cuestiones relativas a la construcción de clientes que soliciten los servicios del modelo implementado. Y por último, en la sección 5 se expondrán las conclusiones del trabajo realizado.

2. Construcción de un Servicio Web WSRF

El WSRF introduce la noción de WS-Resource como base para la construcción de Servicios Grid.

Cuando un WS-Resource es empaquetado como un *Grid ARchive* (GAR) y desplegado en el *container* de GT4, es reconocido por el mismo como un Servicio Web válido que sigue WSRF. Esto es sinónimo de *Servicio Grid*.

Desde el punto de vista del desarrollador, los pasos involucrados para implementar un Servicio Web WSRF para desplegarlo dentro de un *container* de GT4, son los siguientes:

1. **Definir la interfaz del servicio.** Significa preparar el archivo WSDL que define las operaciones del servicio WSRF, y puede incluir definiciones de las propiedades de recursos.

2. **Implementar el servicio.** Se refiere a desarrollar el código fuente para las operaciones del servicio WSRF y propiedades asociadas si las hubiera. En el caso particular del *VisorService*, se eligió el lenguaje de programación *Java* para la implementación del servicio, aunque *Globus* también cuenta con soporte para los lenguajes *C* y *Python*.

3. **Definir los parámetros de despliegue.** Se refiere a preparar un archivo *Web Service Deployment Descriptor* (WSDD) para el servicio WSRF, que define varios aspectos de la configuración del mismo.

4. **Compilar y generar el archivo GAR.** La compilación y creación del archivo GAR involucra la creación de los archivos con los *stubs* apropiados para manejar mensajería SOAP y empaquetar el servicio en un formato requerido por el *container* de GT4.

5. **Desplegar el servicio.** Implica descomprimir el contenido del archivo GAR en una ubicación preestablecida dentro de la estructura de directorios de GT4 de forma que el servicio esté disponible para quien quiera invocarlo.

3. Un Servicio Grid como cliente de otro Servicio Grid

En el resumen de este trabajo se expuso a los Servicios Grid como una herramienta para implementar un control sobre los recursos de un Grid. Con el fin de tener una primera aproximación a esta herramienta, se implementó un modelo basado en Servicios Grid que permita mostrar información de estado de cada uno de los nodos que componen un cluster. Esta prueba fué realizada sobre uno de los cluster pertenecientes al Laboratorio de Investigación en Sistemas Distribuidos (LISiDi) de la Universidad Nacional del Sur (UNS).

Este cluster está formado por nueve computadoras Pentium IV de 3 GHz, con 512 Mb de memoria RAM, discos de 80 Gb y placas de red de 1 Gbps cada una, todas conectadas a un switch. El motivo por el cual se requiere una alta velocidad de comunicación radica en la necesidad de crear la ilusión de una única máquina n veces más potente que una PC común formada por n PCs comunes. Sobre ellas se instaló Globus Toolkit 4 (GT4), que provee el *middleware* necesario para implementar un Grid sobre un cluster de máquinas.

En cuanto al modelo implementado, *VisorService*, se tomó la decisión de dividirlo en dos: un servicio *slave* que se ejecute en cada nodo del cluster y que devuelva, bajo solicitud, la información de estado de la máquina en la que se encuentra; y un servicio *master* que se encuentra alojado solo en la máquina cabecera del cluster y que se encarga de solicitar información de

estado a los servicios slave en cada una de las máquinas con el fin de juntarla en una sola estructura de datos y entregarla a quien solicite el servicio master.

Esto se hizo con el objetivo de ver como dos o más Servicios Grid pueden interactuar entre sí, concientes de que tal vez no es la opción más eficiente debido a que estas solicitudes tienen que atravesar una capa de software correspondiente a GT4 en más de una ocasión. En futuras implementaciones, se reestructurará el servicio de forma que persista el *master* y la información necesaria de todos los nodos sea obtenida mediante una tecnología de más bajo nivel como pueden ser los *sockets* o algún estándar de computación distribuida como MPI.

3.1. Estructura de directorios del servicio

Los archivos que conforman un Servicio Grid, y que han sido mencionados en los pasos de construcción del mismo, deben encontrarse en localizaciones determinadas dentro de una estructura de directorios cuya raíz corresponde al directorio de trabajo del Servicio Grid (el directorio en el que se van guardando los archivos hasta el momento de compilarlos).

Así que como paso "cero" al conjunto de pasos a seguir mencionados en la sección anterior, se establece la jerarquía de directorios necesaria. Dentro de la misma jerarquía pueden estar tanto los archivos pertenecientes al servicio slave como al servicio master.

En este caso, el directorio de trabajo va a ser *VisorService*. Dentro de éste, se va a establecer la siguiente estructura de directorios:

- **schema**: Este directorio debe estar obligatoriamente. Contendrá los archivos de interfaz WSDL necesarios para poder exportar las operaciones de los servicios presentados.
  - *VisorSlaveService*: este subdirectorio de *schema* contendrá el WSDL del servicio slave de VisorService.
  - *VisorMasterService*: de igual manera, contendrá el WSDL correspondiente al servicio master de VisorService.
- **grid/cs/uns/visor**: Estos cuatro directorios se mapearán al nombre de *package* de Java grid.cs.uns.visor. En el directorio *visor*, entonces, se encontrarán las carpetas que contendrán los archivos Java de los servicios, y también la implementación de unos clientes en modo de terminal de texto que serán usados para probar el servicio.
  - *clients*: Esta carpeta, como lo indica su nombre, tendrá los clientes necesarios para probar los servicios. Al igual que la carpeta *schema*, cuenta con los subdirectorios *VisorSlaveService* y *VisorMasterService* para separar los clientes correspondientes a uno y otro servicio, respectivamente.
  - *services*: Como también se puede deducir, este directorio contendrá los archivos de implementación de los servicios.
    - **master** | **slave**: Estos dos subdirectorios tienen una estructura idéntica. Contienen los archivos de despliegue de servicio y un subdirectorio *impl* que contiene los archivos Java que implementan los respectivos servicios.

3.2. Servicio *slave* del VisorService

En esta sección se mostrará cuestiones relativas al cumplimiento de cada uno de los pasos involucrados en el desarrollo de un Servicio Grid. En la subsección 3.3 se mencionarán las

diferencias que se presentan en el desarrollo del servicio master con respecto al del servicio slave.

3.2.1. Definición de la interfaz del servicio

Como primer paso a la construcción del servicio slave, se define la interfaz del servicio: principalmente, qué operaciones va a brindar el servicio sin entrar en cuestiones de implementación y cuáles son los atributos de datos que constituyen su "estado de recurso". Para esto se hace uso de un lenguaje XML especial que puede ser utilizado para especificar qué operaciones ofrece un Servicio Web: el *Web Service Description Language* (WSDL).

Este lenguaje permite abstraerse en esta etapa del lenguaje que se utilizará para implementar las operaciones del servicio.

En el caso particular del *VisorService*, no sería necesario mantener el estado del recurso, debido a que la información que se desea mostrar corresponde a un instante cualquiera y no tiene relación con llamadas anteriores. Sin embargo, dado que es una primera aproximación a la implementación de un Servicio Web, se dividió la tarea de mostrar la información de un nodo en dos operaciones:

- **ejecutar**: que se encarga de recolectar los datos y asignarlos a las propiedades de recursos (atributos que constituyen el estado del recurso).
- **mostrar**: que toma las propiedades de recursos y crea una estructura de datos apropiada para devolver estos datos a quien solicite el servicio.

La estructura de datos que devuelve la operación mostrar (*mostrarResponse*) tiene como atributos los mismos que se utiliza como propiedades de recurso:

- **HostID**: El nombre del nodo (*hostname*) al cual pertenece la información.
- **CpuMHz**: Velocidad del procesador del nodo.
- **CpuUse**: Porcentaje actual de utilización del CPU.
- **MemTotal**: Memoria RAM disponible del nodo.
- **MemPerc**: Porcentaje actual de utilización de memoria sobre la cantidad de memoria RAM disponible (es decir, más de un 100 % indicaría que se está utilizando memoria virtual).

3.2.2. Implementación del servicio

En este punto del desarrollo del servicio, se implementó en Java el código que obtiene los datos de estado del sistema. Java, al ser un lenguaje basado en máquina virtual, no cuenta con operaciones directas que permitan ejecutar comandos del *shell* como podría ser la función `system` del lenguaje C.

Sin embargo, se puede simular a través de las siguientes líneas de código:

```
String cmdString = "..."; // el comando que quiero ejecutar
try
{
    Process p = Runtime.getRuntime().exec(cmdString);
```

```
    BufferedReader pOutput =
    new BufferedReader(
    new InputStreamReader(p.getInputStream()));

    String line;
    StringBuffer tmpCommandOutput = new StringBuffer();

    try
    {
        while ((line = pOutput.readLine()) != null)
            tmpCommandOutput.append(line).append("\n");
        salida = tmpCommandOutput.toString();
    }
    catch (IOException e) {}

    p.waitFor();
    pOutput.close();
}
catch (IOException e) {}
catch (InterruptedException e) {}
```

En esencia, lo que hace este bloque de instrucciones es, en primer lugar, crear un objeto *Process* que representa la ejecución de un comando de *shell* contenido en el string *cmdString*. Luego se crea un buffer a través del cual se va a recibir lo que el comando vuelca en la salida estándar (que visto desde la óptica del programador sería un buffer de entrada porque se quiere leer esta información). Por último, el *try-catch* interno, a través de un bucle, va obteniendo línea por línea la salida del comando desde el buffer y lo va anexando a un *StringBuffer* que, finalmente, se convierte y devuelve como un string (*salida*).

Para ejecutar un comando que provea de la información necesaria, se tuvo que elaborar un *script* con comandos de GNU/Linux que obtuvieran los datos necesarios y filtraran la información de forma que la salida de la ejecución del script contuviera en cada línea un dato a ser asignado directamente a una propiedad de recurso o a ser utilizado para calcular una propiedad de recurso. El *script* debe encontrarse en todas las máquinas que vayan a correr el servicio en el directorio `/usr/bin` de nuestro sistema GNU/Linux.

La implementación del servicio en Java tiene que estar en el subdirectorio grid/cs/uns/visor/services/slave/impl del directorio de trabajo.

3.2.3. Definición de los parámetros de despliegue

Son dos los archivos que deben encontrarse en el subdirectorio grid/cs/uns/visor/slave (el padre del directorio donde se encuentran los archivos Java) para poder desplegar el servicio en el *container* de GT4: `deploy-jndi-config.xml` y `deploy-server.wsdd`.

En el primero, lo que se puede observar es que en la siguiente etiqueta se define el nombre del servicio:

```
<service name="visor/slave/VisorService">
```

es decir, la URI del servicio que va a aparecer al ejecutar el *container* es `http://nombre-dominio-o-IP:8080/wsrf/services/` seguido de lo especificado en la propiedad `name` de la etiqueta `service`. En este caso, sería:

```
http://grid.cs.uns.edu.ar:8080/wsrf/services/visor/slave/VisorService
```

En cuanto al segundo archivo, la siguiente etiqueta produce la asociación entre el servicio y la clase Java que implementa el mismo:

```
<parameter name="className"
value="grid.cs.uns.visor.services.slave.impl.VisorService"/>
```

Otra etiqueta de importancia, es la que define el archivo WSDL que debe utilizarse. Esto es relevante para cuando estamos trabajando con clientes en otras plataformas, ya que es necesario pasarles una URL que devuelva el WSDL con el que podrán crear sus clases o tipos de datos con los que se accederá al servicio.

El WSDL que se utilizará en última instancia no será el que fue editado al principio porque este último contiene órdenes a un preprocesador de WSDL (wsdlpp) que permitirá agregar al WSDL final las líneas necesarias para usar operaciones propias de *Globus* para el establecimiento de referencias al servicio y de la implementación de operaciones comunes a los Servicios Grid desarrollados con GT4 como *GetResourceProperties*.

Es por ello que es necesario indicar el documento que será utilizado para elaborar una respuesta ante la solicitud del archivo WSDL por parte del cliente. Esto se consigue a través de la siguiente etiqueta:

```
<wsdlFile>share/schema/VisorSlaveService/Visor_service.wsdl</wsdlFile>
```

donde `Visor_service.wsdl` es el documento que generará el preprocesador de WSDL (Visor.wsdl es el archivo WSDL que se estableció en el primer paso).

3.2.4. Compilación y generación del archivo GAR

Para simplificar el proceso de compilación se utiliza *globus-build-service*. Esta herramienta, que está incluida como parte del proyecto Globus Service Build Tools (GSBT) [2] de SourceForge, está formada por un archivo *buildfile* de Ant de propósito general y un *script* que dado un conjunto de archivos Java, WSDL, WSDD que respetan una estructura de directorios específica, generará un archivo GAR sin la necesidad de editar manualmente el archivo Ant. Este es el mismo Ant *buildfile* y *script* incluido en el *Globus Toolkit 4 Programmer's Tutorial* [3] en el cual nos basamos para desarrollar este servicio.

Este par de archivos se copian al directorio de trabajo del servicio (*VisorService*) que se denomina como `$BUILD_DIR`. Para que el *buildfile* de Ant de propósito general funcione para el servicio en cuestión es necesario especificar dos parámetros al ejecutar el *script*:

- `SERVICE_DIR` que es el directorio que contiene todos los archivos de implementación y despliegue. En el caso del servicio slave el directorio sería
  grid/cs/uns/visor/services/slave
- `SCHEMA_FILE` que es el archivo WSDL con la descripción de la interfaz del servicio. En este caso, el archivo sería
  schema/VisorSlaveService/Visor.wsdl

Es importante observar que tanto `SCHEMA_FILE` como `SERVICE_DIR` son relativos a `$BUILD_DIR`

El archivo GAR

El archivo GAR es generado en `$BUILD_DIR/GAR_ID.gar`. El `GAR_ID` es generado a partir del parámetro `SERVICE_DIR` reemplazando los separadores de path (/ en UNIX o GNU/Linux) con guiones bajos. Por ejemplo, si el parámetro es:

```
grid/cs/uns/visor/services/slave
```

Entonces, el archivo GAR se generará en:

```
$BUILD_DIR/grid_cs_uns_visor_services_slave.gar
```

El directorio `build`

Todos los archivos intermedios generados por el *script* son ubicados en `$BUILD_DIR/build`. A veces, éste produce resultados inesperados si se realizan muchas compilaciones (el directorio build queda lleno de archivos intermedios de compilaciones previas). Cada vez que se obtenga un error inesperado, la primera cosa que se debería intentar es borrar el directorio build, que asegurará que el build script comience desde cero.

Aunque generalmente no es necesario mirar los contenidos del directorio build, a menudo se quiere verificar si los archivos de *stub* (que permiten la serialización y deserialización de los datos) fueron generados correctamente. Los archivos fuentes de los *stubs* son generados en:

```
$BUILD_DIR/build/stubs-GAR_ID/src
```

Y los *stubs* compilados se ubican en:

```
$BUILD_DIR/build/stubs-GAR_ID/classes
```

3.2.5. El despliegue del servicio

Una vez listo el archivo GAR, se está en condiciones de desplegar el servicio con el comando

```
globus-deploy-gar grid_cs_uns_visor_services_slave.gar
```

Hay tener en cuenta que se necesita especificar el `GAR_ID` (no el nombre de archivo GAR) cuando se hace el *undeploy* (eliminar el servicio de la lista de servicios disponibles) de los archivos GAR usando el comando *globus-undeploy-gar*. Por ejemplo:

```
globus-undeploy-gar grid_cs_uns_visor_services_slave
```

3.3. Servicio *master* del VisorService

Como se había mencionado en la sección 3, el servicio master corre en el nodo cabecera y es el que se va a encargar de solicitar el servicio slave en cada nodo del cluster.

Los pasos para la construcción del mismo son iguales que para el servicio slave, a excepción de que solo contará con una única operación llamada *juntar* que se encarga de solicitar a cada slave que ejecute su servicio y luego recolecta la información que devuelven.

Para lograr la conexión con los servicios slave de cada máquina, el servicio master debe crear una conexión con cada máquina. Para ello es necesario crear una referencia al servicio slave en cada máquina y con esa referencia solicitar las operaciones *ejecutar* y *mostrar*. El código en Java que realiza esta tarea es el siguiente:

```
VisorServiceAddressingLocator locator = new VisorServiceAddressingLocator();
EndpointReferenceType endpoint;

endpoint.setAddress(new Address("http://" + hostID +
":8080/wsrf/services/visor/slave/VisorService"));

// Obtengo el PortType
VisorPortType viz = locator.getVisorPortTypePort(endpoint);

// Llamo a ejecutar
viz.ejecutar(new Ejecutar());
// Ejecutar representa un parametro nulo (necesario para XML Schema)

// Accedo al resultado y lo asigno al recurso correspondiente
MostrarResponse buffer = viz.mostrar(new Mostrar());
// Mostrar tambien representa un parametro nulo para esta operacion.
```

La información devuelta por *juntar* se encuentra en un formato que está definido en el WSDL del servicio slave. Este tipo de objeto devuelto incluye un campo *hostID* que contendrá un *string* para identificar de que máquina proviene (es un nombre de host) o tendrá una cadena `<desconocido>` para indicar que la operación no se concretó debido a que hubo una excepción producida en la conexión (generalmente significa que la máquina está apagada, el servicio no está corriendo o el *container* no está ejecutándose).

Como se mencionaba anteriormente, el formato de este objeto (llamado *MostrarResponse*) se encuentra en el WSDL del slave. Debido a que la información que mantendrá el master es un arreglo con tantas de estas componentes como máquinas, existen dos alternativas para poder usar este formato:

1. Incluir la definición del WSDL del slave en el WSDL del master.
2. Crear un formato exactamente igual (el cual se llamará *status*) y agregar una operación privada que convierta del formato *mostrarResponse* al formato *status*. Dado que esto termina programándose en Java y se puede importar los *stubs* del slave, es posible invocar sus métodos *get* sobre cada una de las propiedades de recurso (coincidentes con los atributos de *mostrarResponse*) para obtener los valores necesarios para utilizar los métodos *set* incluidos en el formato *status*.

La alternativa que se tomó fue la segunda, por comodidad y para no adulterar el archivo WSDL del master con cosas que se podían solucionar desde el lenguaje de programación. Una mejora al servicio sería hacerlo con la primera alternativa, teniendo la ventaja (posiblemente) de que sea más independiente del lenguaje.

4. Implementación de clientes que utilicen el servicio

Una vez que el servicio fué compilado con *globus-build-service*, que fué desplegado en el *container* de GT4 con *globus-deploy-gar* y que la URI del servicio aparece en la lista de servicios al ejecutar *globus-start-container*, está asegurado que el servicio es *sintácticamente* correcto. Ahora se necesita verificar que el servicio realiza la tarea para la cual fué programado. Para

ello es necesario que una aplicación cliente establezca contacto con el servicio y solicite sus operaciones.

Para implementar un cliente de este tipo, se podría hacer un cliente en Java que establezca contacto con el servicio master corriendo en la cabecera del cluster, usando líneas de código similares a las expuestas en la subsección 3.3 cuando se quiso establecer contacto con los servicios slave en cada nodo del cluster.

Sin embargo, el cliente está accediendo desde la misma plataforma de sistema operativo y lenguaje de programación; es decir, el cliente se encuentra en un ambiente homogéneo con el servicio. Incluso está utilizando los *stubs* compilados por el mismo servicio para comunicarse con él. Esta situación no va a ser siempre así, ya que clientes corriendo en otros sistemas operativos, programados en otros lenguajes podrían querer acceder al Servicio Grid (de esto se trata la heterogeneidad, existente en ambientes Grid). Por eso, en la subsección siguiente veremos las cuestiones a considerar para este caso.

4.1. Un cliente creado en otra plataforma

La motivación principal de elaborar el *VisorService* fue la de desarrollar un servicio de información para un sistema de visualización utilizado por el Laboratorio de Investigación en Visualización y Computación Gráfica (VyGLab) que desarrolla sus actividades en el mismo departamento académico que el LISiDi.

El sistema de visualización utiliza la información provista por el *VisorService* para mostrarla de una forma gráfica, intuitiva y fácil de comprender para el ser humano; de manera de poder apreciar de forma rápida y sencilla el balance de carga en un sistema distribuido [4].

Este proyecto en conjunto entre estos dos grupos se viene gestando desde hace tiempo [5, 6] y en *Servicios Grid* se encontró una herramienta para integrar el trabajo de ambos.

La plataforma que utiliza el sistema de visualización utilizado por el VyGLab consta de máquinas con sistema operativo MS Windows XP y Visual Studio .NET como herramienta de programación. Lo cual demuestra la heterogeneidad que se puede presentar al trabajar en un entorno grid. Las clases de *stub* que necesita utilizar el cliente ya no sirven, sino que la plataforma MS Windows/.NET debe crear sus propios *stubs* a partir de la información provista por el archivo WSDL correspondiente al servicio y con ellos realizar las llamadas a los métodos del servicio.

En primer lugar fué necesario cambiar la configuración del *container* de Globus, para que figure *grid.cs.uns.edu.ar* en las URI de los servicios en lugar de la dirección IP correspondiente al nodo cabecera. Para ello, en el siguiente archivo de confugaración

```
$GLOBUS_LOCATION/etc/globus_wsrf_core/server-config.wsdd
```

se debe modificar el valor de `logicalHost` en la siguiente etiqueta:

```
<parameter name="logicalHost" value="grid.cs.uns.edu.ar" />
```

El valor de `logicalHost` puede ser tanto un nombre de host como una dirección de IP. En este caso el nombre *grid.cs.uns.edu.ar* corresponde al host que funciona de cabecera del cluster.

El segundo inconveniente encontrado fué que en el cliente de modo texto sólo se necesitaba la URI del servicio que figuraba en la lista de servicios desplegados en el *container* para poder conectarse al servicio. O sea,

```
http://grid.cs.uns.edu.ar:8080/wsrf/services/visor/master/VisorService
```

Sin embargo, esta URI no era válida para *Visual Studio .NET*, sino que necesitaba la WSDL del servicio. De hecho, al tratar de ingresar esa URI a un *web browser*, emitía el mensaje "File Not Found", a pesar de que el servidor se encontraba escuchando en ese IP y puerto.

El problema es, justamente, que *Visual Studio .NET* espera un descriptor de Servicio Web porque ve al Servicio Grid como tal. Por ende, es necesario agregarle un parámetro a la URI provista de forma que devuelva el archivo WDSL correspondiente. Este parámetro es "wsdl":

```
http://grid.cs.uns.edu.ar:8080/wsrf/services/visor/master/VisorService?wsdl
```

Esta URI se probó sobre un pequeño cliente en modo texto programado en .NET y funcionó de forma satisfactoria, permitiendo crear una referencia a nuestro servicio y realizar subsiguientes llamadas a los métodos del mismo. Sobre esta base, se comenzó el desarrollo, a cargo del VyGLab, de un cliente que utilice como base el sistema de visualización mencionado anteriormente y como herramienta de programación a *Visual Studio .NET*.

5. Conclusiones y trabajos futuros

A través de esta experiencia programando y testeando Servicios Grid, se observa que es una herramienta adecuada (aunque no la más eficiente) para modelar un sistema para el monitoreo de los recursos computacionales en un cluster de computadoras, como prueba a escala de lo que conformaría una infraestructura grid. Alternativamente, se podría utilizar tecnologías pertenecientes al área de *clustering* como pueden ser MOSIX[7], Condor[8], MonALISA[9] que son sistemas que, entre sus funciones, está la de monitorear los recursos en un cluster, y utilizar estos sistemas como proveedores de información para MDS que el sistema de monitoreo provisto por Globus Toolkit 4 y que también está basado en Servicios Grid. Sobre esta alternativa se investigará en el futuro.

Sin embargo, *globus-build-service* constituye una herramienta realmente útil a la hora de generar archivos GAR a partir de un conjunto de archivos Java, WSDL y WSDD que implementan un Servicio Grid sencillo, evitando el trabajo tedioso de aprender a manipular el formato de *buildfile* de Ant y de crear un *builfile* distinto para cada Servicio Grid que se implemente. Sin embargo, su aplicación esta limitada a ejemplos simples y requiere de la colaboración de desarrolladores que puedan extender las funcionalidades de esta herramienta de forma de convertirla en útil para cualquier proyecto que se desee compilar.

Su uso para la compilación del servicio resultó de mucha utilidad y permitió observar los errores que se pueden llegar a presentar ante una mala manipulación de los archivos; además de permitir concentrarse en la implementación del servicio dándole menos relevancia a cuestiones de configuración.

A pesar de ello, existen herramientas que permiten abstraerse aún más de las cuestiones técnicas de los Servicios Grid y que permiten enfocarse en la implementación del servicio, como pueden ser los *plugins* para entornos integrados de desarrollo (IDEs) para programar en Java. Esta opción fue abordada en una primera instancia, sin embargo se abandonó debido a que se presentaron problemas al tratar de compilar y se decidió abordar otra herramienta para programar Servicios Grid.

Sin embargo, las experiencias con *globus-build-service*, al ser una herramienta de más bajo nivel, proveyeron de conocimientos útiles para poder volver a analizar esta alternativa en un futuro, de forma de poder llevar a cabo la tarea de programar un Servicio Grid de forma más automatizada.

Referencias

[1] Miriam Lechner and Martin Chuburu. *Computación Grid, Globus Toolkit y potencia computacional sin límites*, 2006.

[2] *Globus Service Build Tools*. http://gsbt.sourceforge.net/.

[3] Borja Sotomayor. *The Globus Toolkit 4 Programmer's Tutorial*, 2005. http://gdp.globus.org/gt4-tutorial.

[4] Martin Chuburu, Javier Echaiz, and Jorge Ardenghi. Monitoreo de Recursos computacionales en un cluster utilizando Grid Services. *IX Workshop de Investigadores en Ciencias de la Computación, WICC 2007*, pages 612–616, May 2007.

[5] Martín Larrea, Sergio Martig, Silvia Castro, and Javier Echaiz. Visualización del Balance de Carga en un Sistema Distribuido. *11mo Congreso Argentino de Ciencias de la Computación (CACIC 2005)*, pages 1761–1771, October 2005.

[6] Martín Larrea, Sergio Martig, Silvia Castro, and Javier Echaiz. A proposal from the point of view of Information Visualization and Human Computer Interaction for the visualization of distributed system load. *Special Issue on Selected Papers from CACIC 2005, JCS&T*, pages 327–333, December 2005.

[7] Najib A. Kofahi, Saeed Al Zahrani, and Syed Manzoor Hussain. MOSIX evaluation on a linux cluster. *Int. Arab J. Inf. Technol*, 3(1):62–68, 2006.

[8] Douglas Thain, Todd Tannenbaum, and Miron Livny. Distributed computing in practice: the condor experience. *Concurrency - Practice and Experience*, 17(2-4):323–356, 2005.

[9] Harvey B. Newman, I. C. Legrand, Philippe Galvez, R. Voicu, and C. Cirstoiu. MonALISA: A distributed monitoring service architecture. *CoRR*, cs.DC/0306096, 2003.

[10] Martin Chuburu, Miriam Lechner, Javier Echaiz, and Jorge Ardenghi. Experiencias con Globus Toolkit. *VIII Workshop de Investigadores en Ciencias de la Computación, WICC 2006*, pages 217–220, June 2006.

[11] Bart Jacob, Michael Brown, Kentaro Fukui, and Nihar Trivedi. IBM redbook: Introduction to grid computing, 2005.

[12] Ian Foster. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[13] Ian Foster and Carl Kesselman. *The Grid - Blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers, 1999.

[14] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150:1–??, 2001.

Job Scheduling considering Best-Effort and Soft Real-Time Applications on Non-dedicated Clusters

J.García, M.Hanzich, P. Hernández, E. Luque*
Dept. de Informática, Universitat Autònoma de Barcelona
Bellaterra, 08290, Spain
{jrgarcia,mauricio}@caos.uab.es, {emilio.luque,porfidio.hernandez}@uab.es

and

J. Lérida, F.Giné, F.Solsona
Dept. de Informática e Ing. Industrial, Universitat de Lleida
Lleida, 25001, Spain
{jlerida,sisco,francesc}@diei.udl.es

Abstract

As Network Of Workstations (NOWs) emerge as a viable platform for a wide range of workloads, new scheduling approaches are needed to allocate the collection of resources from competing applications. New workload types introduce high uncertainty into the predictability of the system, hindering the applicability of the job scheduling strategies. A new kind of parallel applications has appeared in business or scientific domains, namely Soft Real-Time (SRT). They, together with new SRT desktop applications, turn prediction into a more difficult goal by adding inherent complexity to estimation procedures. In previous work, we introduced an estimation engine into our job scheduling system, termed CISNE. In this work, the estimation engine is extended, by adding two new kernels, both SRT aware. Experimental results confirm the better performance of simulated respect to the analytical kernels and show a maximum average prediction error deviation of 20%.

Keywords: parallel processing, soft real-time, job scheduling, non-dedicated clusters.

Resumen

Mientras las Redes de Estaciones de Trabajo (NOWs) emergen como una plataforma viable para un amplio espectro de aplicaciones, son necesarios nuevos enfoques para planificar los recursos disponibles entre las aplicaciones que compiten por ellos. Los nuevos tipos de cargas introducen una alta incertidumbre en la predictibilidad del sistema, afectando la aplicabilidad de las estrategias de planificación de tareas. Un nuevo tipo de aplicaciones paralelas, denominado tiempo real débil (SRT), ha aparecido tanto en los ámbitos comerciales como científicos. Las nuevas aplicaciones paralelas SRT, conjuntamente con los nuevos tipos de aplicaciones SRT de escritorio, convierten la predicción en una meta aún más difícil, al agregar complejidad a los procedimientos de estimación. En trabajos anteriores dotamos al sistema CISNE de un motor de estimación. En este trabajo añadimos al sistema de predicción fuera de línea dos nuevos núcleos de estimación con capacidad SRT. Los resultados experimentales muestran un mejor rendimiento del núcleo simulado con respecto a su homólogo analítico, mostrando un promedio de desviación máximo del 20%.

Keywords: procesamiento paralelo, tiempo real débil, planificación de tareas, clusters no dedicados.

*This work was supported by the MEyC-Spain under contract TIN 2004-03388.

1 INTRODUCTION

Nowadays, wasted computational resources are a common reality in open laboratories using NOWs in any university. The aim is to take advantage of those available resources to do parallel computation [1]. The possibility of using this computing power to execute distributed applications with a performance equivalent to a Massively Parallel Processor (MPP) and without perturbing the local user's applications performance has led to proposals for new resource management environments [7]. As a try to take advantage of these idle computational resources, these environments combine space sharing and time sharing scheduling techniques. In order to provide such a system, we developed CISNE [7].

CISNE is composed basically of a dynamic coscheduling technique and a space sharing scheduler. The coscheduling technique [5] ensures the progress of running parallel jobs without disturbing the local users, even when using a Multiprogramming Parallel Level (MPL) greater than one [5]. The space sharing scheduler is named LoRaS and is component of CISNE that performs the parallel workload distribution among the cluster nodes. LoRaS takes into account the state of the cluster and the characteristics of the local and parallel workload to implement different policies used in job allocation. To achieve better scheduling decisions, CISNE needs to foresee the state of the cluster. Likewise, this prediction capacity could help to guarantee some limits in the turnaround and missdeadline time to the applications of any user.

Unfortunately, new desktop SRT applications [4] turn prediction into a more difficult goal, introducing great uncertainty into the predictability of the system. A situation that gets worse with the new kind of parallel SRT applications that has arisen in business and scientific domains [13]. This new kind of parallel applications usually requires bounded response time or guaranteed turnaround time [9], [12]. In this context, it is mandatory to ensure that Best-effort applications do not affect the deadline requirements for the SRT jobs, and to protect the Best-effort applications from starvation.

According to this new situation, we are interested in researching new estimation methods, which take Best-effort and SRT applications, local and parallel, into account. Following studies like [8], [11], the choice was a simulation approach to represent our scheduling system. These systems have an estimation kernel, which is in charge of the prediction of the execution time. Unlike previous works, where these kernels are based on the use of a historical repository, we bet by the use of estimation kernels based on simulation or analytical methods. Both proposals were evaluated experimentally. Our results reveal as the simulated kernel achieves better results due to its high degree of detail.

The outline of this work is as follows, section 2 introduces the SRT application framework used in this study, also places our work in the job scheduling area. Section 3 depicts the off-line simulation process in the CISNE, and section 4 describes the new SRT aware approaches for calculating the Remaining Execution Time (**RExT**). Next, section 5 analyzes the experimental results. Finally, the conclusions and the future work are explained.

2 BACKGROUND

This section introduces the novel application framework studied in this paper. Likewise, this also locates our work in the job scheduling area.

2.1 Local and Parallel SRT Applications

An extensive work has been done in Real Time (RT) support [10], mainly directed towards industrial environment. Recently, this knowledge has been gradually migrated to the commercial world. Most of researches efforts focuses in the idea that common operating systems performs badly in presence of applications requiring some kind of resource's priority, usually multimedia applications, such as videos [3], [14], [15].
A common definition of a SRT application is a RT application that may loose some deadlines and no disaster will occur. In the local user environment, a multimedia application suits this definition quite well. It means that nothing happens if a video skips a few frames. The same idea fits for parallel SRT applications. Some users want its results within a deadline but it is not always feasible, maybe a system overload would interfere and cause a missdeadline. However, these undesirable situations should be minimized. According to this aim, neither local nor parallel SRT applications can be ignored. Thus, we will try to fulfill their requirements in the best manner.
Local SRT applications may be *periodic* or *aperiodic* [10]. The periodic tasks are those that are released regularly in periods and must be executed before a certain deadline smaller than the period. Aperiodic tasks are those that are activated irregularly at some unknown and maybe unbounded rate.
From now on, parallel SRT applications will be denoted as par_SRT and local SRT applications as local_SRT.

2.2 Job Scheduling and SRT Applications

In [13], new types of par_SRT are considered and their behavior studied for dedicated clusters by using simulation. Given that the cluster is dedicated, any kind of local load is considered. Regarding to the parallel SRT applications, RT vision applications are taken into account.
Studies like [9], [12] are focused in new types of parallel SRT application, both working in dedicated *Beowulf* clusters. The type of par_SRT considered are RT constrained vision applications in [12] and sensors collecting large amounts of data in [9].
Our work is focused in a novel research area, considering non-dedicated clusters. It means that parallel applications, which can be either par_SRT or not, together with local workload, which can also be either Best-effort or SRT, are running all together in the same cluster, sharing the same computational resources.

3 ENVIRONMENT CONSIDERATIONS

In order to implement and evaluate our estimation proposals, we need a scheduling system oriented towards non-dedicated environments. With this goal, in previous works we developed the CISNE system [7] as an integral scheduling environment that merges both Time and Space sharing subsystems. This section describes this system to achieve a better understanding of the off-line simulation process.

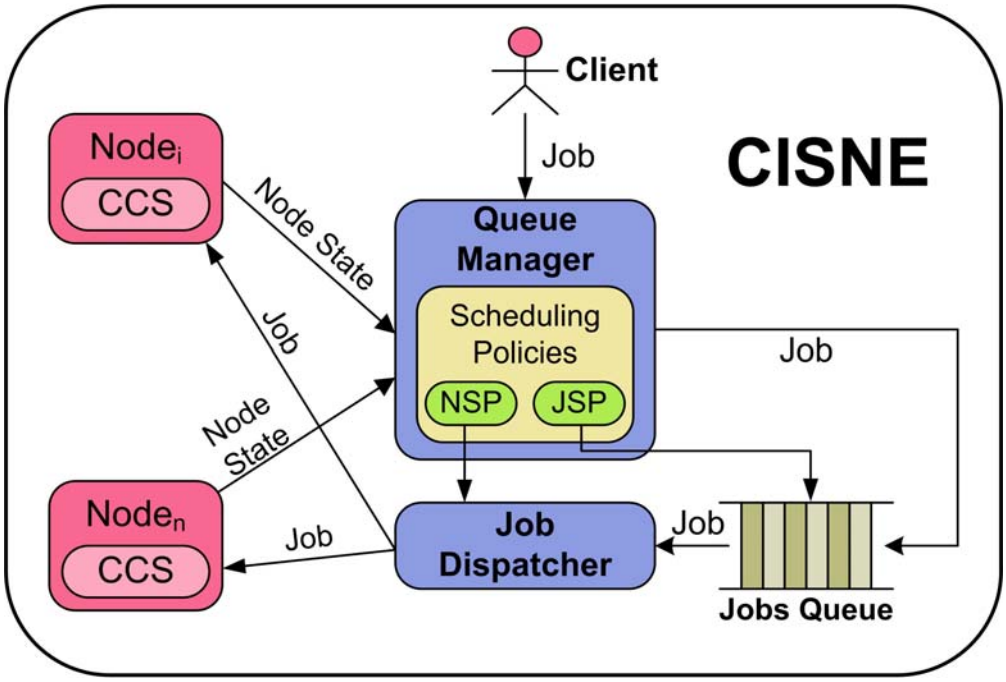


Figure 1. General architecture of the CISNE system.

3.1 Description

The CISNE architecture is depicted in Figure 1. Its main aim is to manage parallel applications in a non-dedicated environment, ensuring benefits for the parallel applications, while preserving the local task responsiveness. When a parallel job is submitted to the CISNE system, the job waits in a queue until the *Queues Manager* decides to schedule it. This decision is taken according to the computational requirements of each parallel job waiting in the queue, together with each node state information received. According to the above description, the CISNE system needs to define the following policies:

- *Job Selection Policy* (JSP) is the policy for selecting the next job to run from the jobs queue. This could depend on the order in the queue (i.e. First-Come-First-Served, *FCFS*), and the estimated cluster state (intrusion level into the local workload, the MultiProgramming Level (MPL) of parallel applications, the memory and CPU usage on each node; i.e. Backfilling). In order to make easy the comprehension of the experimental results, a simple FCFS policy is used through this work.

- *Node Selection Policy* (NSP) is the policy for distributing the parallel tasks among the nodes. This depends on the cluster state and the parallel job characteristics. In this work, we use a policy, named *Normal*, which selects the nodes for executing a parallel application considering only the resources usage level throughout the cluster. This policy does not overload any node in detriment of the local user interactiveness. To achieve this, it establishes an acceptable system usage limit for CPU and main memory by means of a social contract [2] between local and parallel users.

In order to evaluate different workloads and scheduling policies over several simulated cluster environments, an off-line simulation system was included in the CISNE system. Next subsection describes the off-line simulation engine of CISNE.

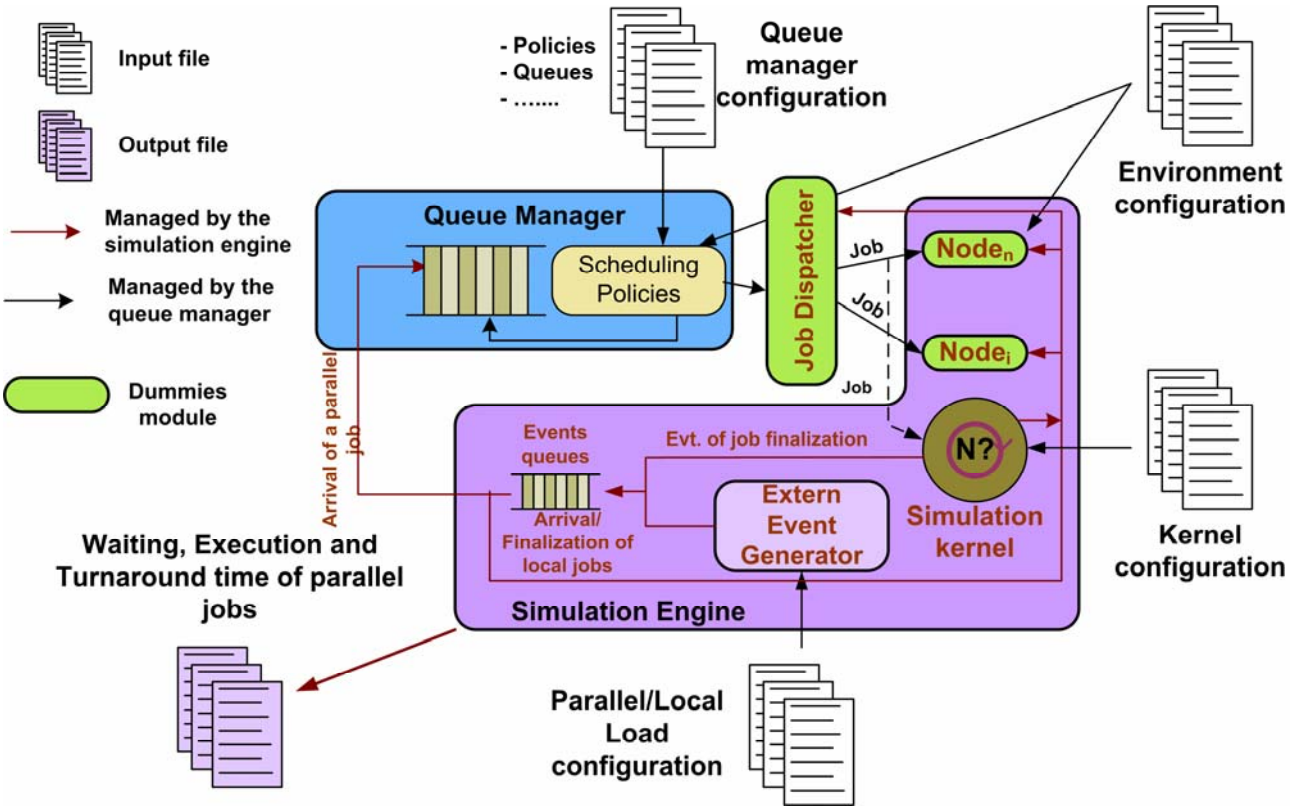


Figure 2. Architecture of the Off-line simulator.

3.2 Off-line Simulation

According to Figure 2, the architecture of the simulator is made up of two main modules. The *Queue Manager*, who manages the Waiting Queue of the system, which means selecting the next job to run and distributing its tasks among the nodes in the cluster. The other main module is the *Simulation Engine*. As shown in Figure 2, our simulator is directed by discrete events, such as the arrival, scheduling or finalization of any kind of jobs. In order to be able to simulate, the Simulation Engine has three different input configuration files:

- *Environment Configuration*: specifies the set of nodes used by the Queue Manager to execute the parallel workload. It describes the number of nodes in the simulated environment and their main characteristics: computational power, memory size and the local load resource usage.
- *Kernel Configuration*: allows the method used to estimate the turnaround time of each parallel job managed by the Queue Manager to be set. It can be either the analytical or simulated estimation method, described in next section.
- *Load Configuration*: allows specification of the list of parallel and local jobs to be simulated. By each job, our simulator needs detailed information, for example: job size, execution time in isolation, arrival time and its maximum requirements of CPU and memory.

Once all the configuration files are loaded, CISNE is ready to start the simulation process, using the application information collected before from real executions. This information, together with the cluster state, is used to generate data sets that conforms the different stages of simulation. A new stage begins with each arrival of a new parallel application. In such a moment, a data set containing current cluster state is created and the new parallel application is included in it.

These kernels, represented by *Simulation kernel* in Figure 2, must return an estimation of the Remaining Execution Time (**RExT**) for the running applications. From the RExT value, our simulator is able to obtain the Turnaround time of each parallel application. It is worth pointing out that the off-line simulation method has been validated in [6],[7].

4 SRT CAPABLE KERNELS ADDED

In order to provide SRT capabilities to the off-line simulation of CISNE, two new estimation kernels were developed, one analytical and the other simulated. These new kernels are described in this section.

4.1 Analytical Method

To make easier to understand our SRT aware analytical kernel, termed $RExT_{ANL-SRT}(j)$, we will describe a non-SRT capable analytical kernel used as seed.

The non-SRT capable method, termed $RExT_{CPU}(j)$ starts by calculating the RExT that the application would need if it were executed in isolation ($RExT_{isol}(j)$). This value is calculated as follows:

$$RExT_{isol}(j) = \frac{t_{tot}(j) \times (t_{Tcpu}(j) - t_{Ucpu}(j))}{t_{Ucpu}(j)}, \quad (1)$$

where $t_{tot}(j)$ is the total execution time, $t_{Tcpu}(j)$ is the amount of CPU time by parallel application *j* running in isolation and $t_{Ucpu}(j)$ is the amount of CPU time used by parallel application *j* from its beginning, a value provided by the *Queue Manager* module.

Note that equation 1 assumes that the $RExT_{isol}(j)$ is proportional to the total execution time in isolation ($t_{tot}(j)$) bounded by the CPU time to be consumed ($t_{Tcpu}(j) - t_{Ucpu}(j)$), and the total amount of CPU time needed by the application ($t_{Tcpu}(j)$).

Next step in $RExT_{CPU}(j)$ method is to consider the CPU requirements of the tasks (in percentage). According to this, the $RExT(j)$ is calculated as follows:

$$RExT_{CPU}(j) = RExT_{isol}(j) \times \frac{CPU(j)}{CPU_{feasible}(j)}, \tag{2}$$

where $CPU(j)$ is CPU percentage ($t_{Tcpu}(j)/t_{tot}(j)$) that the application can use and

$$CPU_{feasible}(j) = \min(CPU(j), \frac{CPU(j)}{CPU_{\max}(j)}) \tag{3}$$

is the maximum CPU usage (in percentage) that we expect the application *j* could use. Finally

$$CPU_{\max}(n) = \max(CPU_{par}(n) + CPU_{loc}(n) \mid n \in N(j)) \tag{4}$$

where $CPU_{loc/par}(n)$ is the sum of the CPU usage of each local/parallel task running in the node *n*. Note that this represents the maximum CPU usage requirements (in percentage) among the nodes where the application *j* is running.

Starting from the $RExT_{CPU}(j)$, we describe our SRT aware analytical proposal, which considers not only the CPU usage, but distinguishes which of the requirements of the parallel and local tasks are SRT. In order to do this, we redefine the $CPU_{feasible}(j)$ expression of $RExT_{CPU}(j)$ method as follows:

$$CPU_{feasibleSRT}(j) = \begin{cases} CPU(j) & j \in S_{srt} \\ \min(CPU(j), CPU_{mS}(j)) & j \notin S_{srt} \end{cases} \tag{5}$$

where S_{srt} is the set of parallel SRT applications currently running in the cluster and $CPU_{mS}(j)$ is defined as

$$CPU_{mS}(j) = \min(\frac{CPU(j) \times (100 - CPU_{SRT}(n))}{CPU_{nonSRT}(n)} \mid n \in N(j)) \tag{6}$$

where $CPU_{SRT}(n)$ and $CPU_{nonSRT}(n)$ are the sum of CPU required by every SRT and non-SRT tasks, respectively, running in the node *n*. The tasks may be either parallel or local.

4.2 The Simulated Approach

Our second SRT aware proposal to compute RExT, which is named $RExT_{SIM-SRT}(j)$, is based on simulation. An external simulator, named *Simulation Kernel* in Figure 2, does the simulation of the successive stages generated by the CISNE estimation system.
Like the analytical estimation method, it takes a snapshot of a simulation point but performs a detailed simulation instead of analytical calculations. $RExT_{SIM-SRT}(j)$ method may assign SRT tasks priorities using several policies, Rate Monotonic Scheduling (RMS [10]) is the policy used in this study. Non-SRT tasks priorities (parallel or local) are assigned using Round Robin.
$RExT_{SIM-SRT}(j)$ method is in fact a whole simulation engine, capable of performing a low level simulation of each node behavior, taking into account available resources and allocation policies. The communication between the two simulation engines, the off-line simulator and the simulation kernel (see Figure 2), is achieved through XML format files containing cluster state and simulation results. In each stage, the simulation kernel takes the cluster state as input and returns the RExT of each parallel job to the off-line simulator.

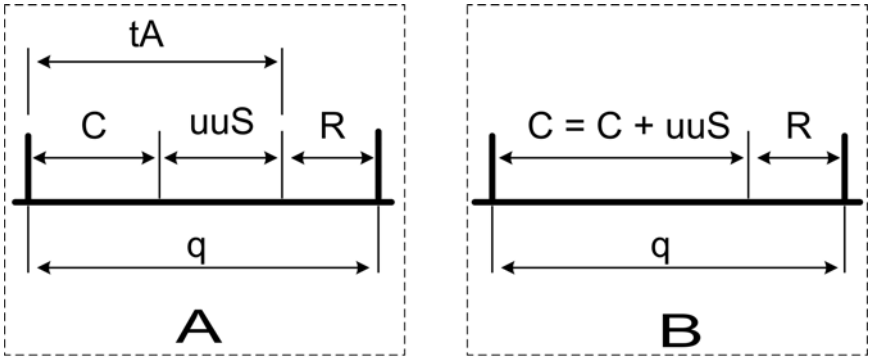


Figure 3. Dynamic slice in per-stage simulator.

Due to space restrictions, only the way $RExT_{SIM-SRT}(j)$ method deals with local_SRT and par_SRT jobs is described more deeply. To simplify our model and be able to predict in our system, we convert par_SRT deadline (*D(j)*) into a period ($T_{par}(j,n)$) and a computation time ($C_{par}(j,n)$). The period is set as a default value, and the computation time is calculated using equation 7.

$$CPU_{mS}(j) = \min\left(\frac{CPU(j)\times(100-CPU_{SRT}(n))}{CPU_{nonSRT}(n)} \mid n \in N(j)\right) \quad (7)$$

where $t_{exec}(j)$ is the current running time for par_SRT *j*. This equation represents a way to ensure that the par_SRT will finish its execution at least within the desired deadline, achieved by reserving to par_SRT the minimal quantum slice it needs.

This is a pessimistic approach and causes the par_SRT to finish very close to its deadline, no matter the cluster load. In order to palliate this effect, the computation time assigned to par_SRT of each CPU quantum is assigned dynamically as the bigger value of the maximum quantum slice available to the par_SRT and the quantum slice needed by the par_SRT according to equation 7.
The dynamic slice process is depicted in Figure 3, where **q** is the CPU quantum, **C** is the reserved computation time to a par_SRT, **R** is the reserved computation slice of any other SRT applications, local or parallel, **uuS** is the unused slice of the quantum and the **tA** represents the available slice to par_SRT. Whenever is possible our simulator assigns all available quantum slice to par_SRT (represented by the **tA** in Figure 3.A), taking into account the CPU needs of other SRT jobs present in the node, local or parallel. Figure 3.B shows a dynamically assigned computation time, which is the sum of **uuS** and **C**, i.e: **tA**. In this new approach, the value calculated using equation 7, **C** in Figure 3, is used as the minimal computation time that the par_SRT must receive. This way a par_SRT may take advantage of low CPU load in a node at any moment.
The comparative values of our two SRT aware estimation kernels described above are shown in the next section.

5 EXPERIMENTATION

In order to carry out the experimentation process, we need two different kinds of workload. On one hand, we need to simulate the local user activity and, on the other, we need some parallel applications arriving at a representative interval.

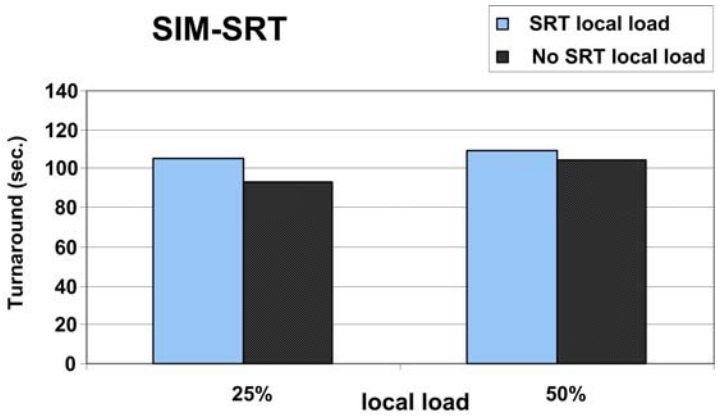


Figure 4. Influence of the local SRT load into the parallel applications performance measured for the $RExT_{SIM-SRT}(j)$ method.

The local user activity is modeled by a benchmark that could be parametrized in such a way that it uses a percentage of CPU, memory and network; which represents the Best-Effort applications. To parametrize this benchmark realistically, we measure our open laboratories for a couple of weeks and use the collected values to run the benchmark (15% CPU, 35% Mem., 0,5KB/sec LAN). We used the CPU (11% or 41%) and memory (15%-20%) requirements of the Xine video player for different visualization window sizes [4] to represent the SRT local workload.

The parallel workload was a list of 90 PVM NAS parallel jobs (CG, IS, MG, BT) previously characterized in our system with a size of up to 16 tasks, which reached the system following a Poisson distribution. These jobs were merged so that the entire workload had a balanced requirement for computation and communication. It is important to mention that the *MPL* reached for the workload depends on the system state at each moment, but in no case will exceed an *MPL*=4 [6]. In this case, and to represent a parallel SRT workload we used the same parameters as for the non-RT parallel applications (execution time in isolation, memory usage, CPU consumption, etc.), but defining a deadline for some of the parallel applications contained in the workload.

5.1 Results

The Figures 4 and 5 shows the influence of the local SRT load on the performance of the parallel applications for both SRT capable estimation methods. All simulation was carried out for 16 nodes. As can be observed there is a difference of near 15% between the best case ($RExT_{SIM-SRT}(j)$ estimation method with 50% of the nodes with some local SRT load, and the worst case ($RExT_{SIM-SRT}(j)$ estimation method with 25% of the nodes with some local SRT load). As expected, turnaround variations depend not only on the estimation method, but also on the local load. A consideration that should be taken into account is the minor turnaround (14% lower in average) returned by the simulation estimation method (Figure 4) compared with the analytical (Figure 5). That effect combined with the fact that $RExT_{SIM-SRT}(j)$ is a pessimistic estimation method, allows us to assume that this prediction is closer to a real execution.

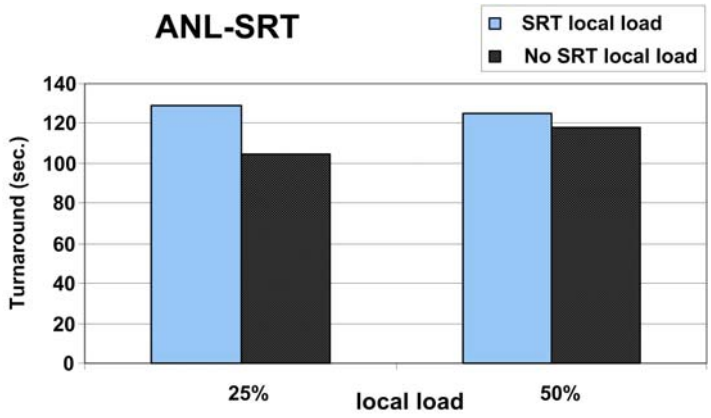


Figure 5. Influence of the local SRT load into the parallel applications performance measured for the $RExT_{ANL-SRT}(j)$ method.

In other experiment we also add parallel SRT applications to the workload for generating Figure 6, where we measure the *Failed Turnaround Percent* of the applications for our two SRT estimation methods. This metric is calculated as the percent of par_SRT that miss theirs turnaround times when considered SRT.

From the Figure 6, it is clear that the simulated method ($RExT_{SIM-SRT}(j)$) performs much better than the analytical one, almost always staying below 20% of missdeadline for the applications.

When local load is considerable (50% a half of which is SRT) and some parallel applications are SRT (30%), the analytical model gives an awfully bad result. The problem relies in the simplicity of the analytical model to estimate a system as complex as a non-dedicated cluster.

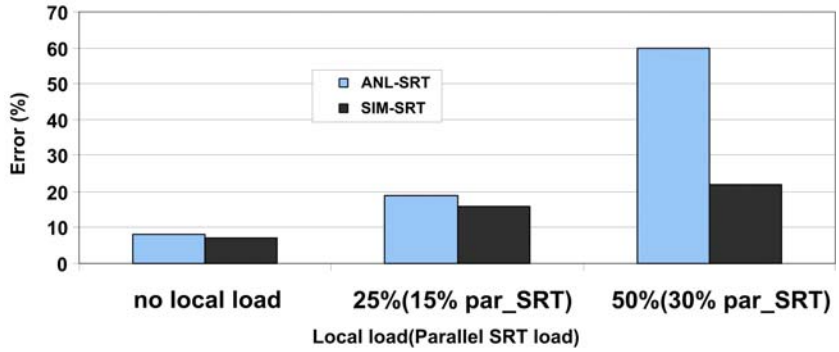


Figure 6. Failed turnaround percent for both SRT estimation methods.

6 CONCLUSIONS AND FUTURE WORK

Due to recent changes in the parallel and local application characteristics, it is urgent to enhance the prediction capacity of a resource management environment over a non-dedicated cluster, to include some sort of SRT estimation schema. In this work we have proposed estimation methods based on analytical and simulation approaches. The main aim of these methods is the prediction of RExT for a given running application and a defined cluster state. Based on such estimation, the system is then capable of estimating the turnaround time of the parallel applications as a parallel user metric.

The results shown in the present work demonstrate that an analytical approach to make some predictions in a non-dedicated SRT environment is possible, as long as local user and par_SRT presence remains low. A simulation approach for making such estimations was shown to be a very good option, even in the presence of high parallel and local SRT loads.

Future work will focus on consider the influence of multi-core processors and a way to take advantage of them in job scheduling. The need to perform exhaustive feasibility analyses of the SRT load is also an open line.

REFERENCES

[1] Acharya, A. and Setia, S. *Availability and Utility of Idle Memory in Workstation Clusters*, Proceedings of the ACM SIGMETRICS, 35-46, 1999.

[2] Arpaci, R. et al. *The Interaction of Parallel and Sequential Workloads on a Network of Workstations*, Proceedings of the ACM SIGMETRICS 1995, 267-277, 1995.

[3] Childs, S. and Ingram, D. *The Linux-SRT Integrated Multimedia Operating System: Bringing QoS to the Desktop*, IEEE Computer Society, 00, 0135, 2001.

[4] Etsion, Y.; Tsafrir, D. and Feitelson, D. G. *Desktop Scheduling: How Can We Know What the User Wants?*, ACM Trans. Multimedia Comput. Commun. Appl., ACM Press, 2, 318-342, 2006.

[5] Gin F. *Cooperating Coscheduling: a coscheduling proposal for non-dedicated, multiprogrammed clusters*, Ph.D. Thesis, Universitat Autónoma de Barcelona, 2004.

[6] Hanzich, M. et al. *Using Simulation, Historical and Hybrid Estimation Systems for Enhancing Job Scheduling on NOWs*, Proceedings of the IEEE International Conference on Cluster Computing, pp 1-12, 2006.

[7] Hanzich, M. et al. *CISNE: A New Integral Approach for Scheduling Parallel Applications on Non-Dedicated Clusters*, *EuroPar 2005*, Lecture Notes in Computer Science, 3648, 220-230, 2005.

[8] Li, H.; Groep, D.; Templon, J. and Wolters, L. *Predicting Job Start Times on Clusters*, Proceedings of CCGrid2004, IEEE Computer Society Press, 2004.

[9] Plale, B.; Turner, G. and Sharma, *A. Real Time Response to Streaming Data on Linux Clusters*, Computer Science Department Technical Report TR-569. Indiana University, 2002.

[10] Sha, L. et al. *Real Time scheduling Theory: A Historical Perspective*, Real-Time Systems, 28, 101-155, 2004.

[11] Smith, W. and Wong, P. *Resource Selection Using Execution and Queue Wait Time Predictions*, NAS Technical Reports, 2002.

[12] Yang, M. et al. *An automatic scheduler for real-time vision applications*, Proceedings of Parallel and Distributed Processing Symposium, 2001.

[13] Zhan, Y. and Sivasubramaniam, A. *Scheduling Best-Effort and Real-Time Pipelined Applications on Time-Shared Clusters*, Proceedings of SPAA'2001, 209-218, 2001.

[14] Hide, E.; Stack, T.; Regehr, J. and Lepreau, J. *Dynamic CPU management for real-time, middleware-based systems*. Proceedings of RTAS 2004. 10th IEEE, 286-295, 2004.

[15] Nieh, J. and Lam, M. *A SMART Scheduler for Multimedia Applications*, ACM Transactions on Computer Systems, 21, 117-163, 2003.

[16] Etsion, Y.; Tsafrir, D. and Feitelson, D. G. *Process prioritization using output production: Scheduling for multimedia*, ACM Trans. Multimedia Comput. Commun. Appl., ACM Press, 2, 318-342, 2006.

Planificación dinámica de clusters a demanda en entornos Grid

Bertogna, Leandro
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue
mlbertog@uncoma.edu.ar

Naiuf Marcelo, De Giusti Armando
Instituto de Investigación en Informática LIDI
Universidad Nacional de La Plata
{degiusti,mnaiouf }@lidi.info.unlp.edu.ar

Resumen
Debido a la gran cantidad de recursos de hardware y software que componen los sistemas Grid, cada uno con diferentes características y complejidades, se torna imperioso simplificar y automatizar su administración. En este trabajo como un primer paso se presentan dos niveles de planificación, el primero a nivel de metaorganización, con el objetivo de generar clusters a demanda basado en requerimientos de aplicaciones y el segundo a nivel de organización local, gestionando los recursos del cluster en forma dinámica para lograr un máximo aprovechamiento de los recursos ofrecidos. Como recursos se estudian máquinas virtuales que interconectadas forman un cluster homogéneo entre organizaciones.

Palabras claves: Grid, planificación, máquinas virtuales

1. Introducción

Debido a la gran cantidad de recursos de hardware y software que componen los sistemas Grid, cada uno con diferentes características y complejidades, se torna imperioso simplificar y automatizar su administración. Tareas de gestión rutinarias que requieren recursos humanos especializados deberían disminuir en forma considerable para transformarse solo en definición de políticas, que sistemas autónomos de gestión de infraestructura cumplieran o hicieran cumplir.

Un punto clave para la gestión de recursos en los sistemas Grid se encuentra en la capa intermedia. Esta coordina recursos e intercambia información sin importar su ubicación física. Cuando alguna aplicación utiliza Grid y realiza requerimientos de hardware para su ejecución esta capa es la encargada de buscar, seleccionar e instanciar los recursos apropiados en donde realizar su despliegue.

La componente que interviene de forma activa de este proceso es el planificador de Grid o metaplanificador. Este permite acceder a los recursos distribuidos en distintas organizaciones físicas y comunicarse con los distintos administradores de recursos locales. Los administradores de recursos se convierten de esta manera en los proveedores de servicio y el metaplanificador los coordina y utiliza, en base a criterios preestablecidos.

Otro aspecto a tener en cuenta es la gestión de los recursos de cada organización física. Aún persisten problemas cuando distintos requerimientos compiten por los mismos recursos dentro de la organización virtual. Poder garantizar un correcto control de la utilización de recursos y su disponibilidad en un entorno adecuado para las aplicaciones es una tarea difícil y generalmente no llevarlo adelante en forma automática conlleva a un uso incorrecto o bajo aprovechamiento.

Una de las alternativa para maximizar la utilización de recursos, es la implementación de máquinas virtuales. Las máquinas virtuales ofrecen la posibilidad de instanciar entornos de trabajo preconfigurados e independientes, tiene la capacidad de administrar y limitar el uso de procesadores, memoria y disco, además de la capacidad de migrar a otra máquina física el entorno completo si fuera necesario. Las implementaciones actuales de máquinas virtuales proveen una performance similar a la obtenida por los sistemas físicos.

La contribución de este trabajo se encuentra en la extensión e implementación de un metaplanificador y un administrador de recursos, en este caso los recursos son clusters de máquinas virtuales. El metaplanificador según el requerimiento de una aplicación y a través de heurísticas de manera automática selecciona equipos físicos del conjunto de servidores disponibles en el Grid y los conecta en una red virtual. El administrador de recursos local en una organización física monitorea y adapta en forma dinámica la carga de trabajo sobre los equipos físicos para optimizar el uso de los mismos.

En la segunda sección se analizarán trabajos relacionados de otros grupos de investigación, en la tercera se describirá la arquitectura donde está incluido el metaplanificador, en las tercer y cuarta sección se describe la extensión del planificador y los algoritmos que lo implementan y por último, se analiza el administrador de recursos local y se realizan las conclusiones.

2. Trabajos Relacionados

Existen diferentes trabajos relacionados con esta problemática tomando como caso de uso entornos de altas prestaciones, cada uno enfatizando diferentes aspectos de la solución. El proyecto Virtual Workspaces[1] pone mayor importancia en la definición del espacio virtual dentro de un entorno Grid. Su caso de uso son los clusters de máquinas virtuales dentro de una red local. Cluster on Demand[2] implementa el empaquetado de un planificador de cluster para obtener subconjuntos de un cluster físico a través de la asignación dinámica de direcciones de red. El proyecto VioCluster[3] se relaciona en gran medida con este trabajo, salvo por la diferencia de que no hace mención a Grid en la configuración de las redes virtuales, o descubrimiento dinámico de máquinas candidatas para instanciar máquinas virtuales; sino que hace hincapié en la negociación automática de dominios de administración según políticas preestablecidas relacionados con conceptos autonómicos. Por último, el proyecto In-VIGO[4] donde el nivel de abstracción es mucho mayor, permite a las aplicaciones hacer uso de entornos virtuales a través de servicios Grid.

3. Arquitectura

La arquitectura donde se inscribe el presente trabajo tiene por objetivo la creación de redes de recursos virtuales dentro de entornos Grid[5]. Los usuarios pueden acceder a los recursos en forma interactiva a través de interfaces web. Se busca realizar una configuración en forma segura y con mínima intervención de los administradores locales en cada organización física.

Desde el punto de vista de diseño, la arquitectura se divide conceptualmente en tres capas. En la primera llamada capa de acceso, los clientes ingresan al sistema, la segunda es la capa de gestión, que controla el acceso y la creación de los recursos definidos en el sistema, y finalmente la capa de recursos que se relaciona con la instanciación de los recursos, en este caso máquinas virtuales sobre hosts físicos.

Los clientes accesan al servicio de planificación del sistema expresando sus requerimientos a través de un lenguaje basado en XML, que en forma paramétrica determinan cuantos nodos virtuales, que imágenes y cuales son los requerimientos de ejecución de la aplicación deseada.

Los nodos Grid cuentan con información del estado de los hosts de la planta física. El planificador, a través del modelo que se presenta en este trabajo, basándose en información de configuración y estado de esos hosts físicos, determina la asignación de nodos virtuales a nodos físicos. Si se requirieran más máquinas virtuales de las que un nodo Grid pudiera ofrecer, por no encontrarse disponibles, o debido a su nivel de utilización, se buscará instanciar recursos virtuales en distintos nodos Grid formando una red virtual entre ellos.

Para el caso de uso de cómputo paralelo, caso este específico para el presente trabajo, la asignación llevada a cabo por el planificador instanciará no más de una máquina virtual por host físico; todas estas máquinas virtuales se encontrarán conectadas en un solo espacio de direcciones; y el resultado devuelto al usuario será un punto de acceso único al conjunto de recursos de la organización.

4 Metaplanificadores

La composición de módulos del metaplanificador se basa en el framework propuesto por la RFC 2753[6], este framework define protocolos y una jerarquía de módulos para el control, basado en políticas. Sus componentes arquitecturales principales son el punto donde las políticas son aplicadas (Policy Enforcement Point - PEP) y donde se toma la decisión de cuales aplicar (Policy Decision Point - PDP). Para el metaplanificador propuesto en este trabajo se ha invertido el flujo de control. En el caso de la RFC los eventos ocurren en los PEP y estos generan requerimientos al PDP para la toma de decisiones, en el caso propuesto en este trabajo, contrariamente la secuencia de eventos comienza con el ingreso de un requerimiento de espacio virtual en el PDP, este consulta el estado a los niveles inferiores y dependiendo de la información obtenida, genera una configuración para satisfacer el requerimiento.

Para permitir que estos puntos de decisión trabajen correctamente y la solución sea escalable se utiliza el principio de alcance en la toma de decisión, minimizando que requerimientos de las organizaciones locales y sus políticas suban a niveles superiores de la jerarquía en la Grid, para lo que se utiliza un punto de decisión local (Local policy desicion point LPDP), extensión prevista en la RFC. La RFC interpreta este LPDP como un representante local del PDP, en esta implementación se sigue la misma filosofía pero orientando su inteligencia a la solución de problemas locales y como representante de la organización local para negociar acuerdos de servicio con el PDP del Grid y hacerlos cumplir.

Para la implementación en primer término se realizó el estudio de distintos metaplanificadores disponibles en el mercado eligiéndose a CSF (Community Scheduler Framework -CSF)[7]. Por su característica, permite reutilizar componentes implementados y extenderlo agregando nuevas implementaciones de acuerdo con el modelo de gestión deseado. Este metaplanificador es una implementación *open-source* que consta de distintos servicios, un sistema de colas con mecanismos de planificación adaptables y extensibles, un servicio de tareas y un servicio de reserva, así como también una estructura para el flujo de información desde los nodos y un sistema de adaptación de las tareas a distintos planificadores en cada uno de los recursos físicos. En este caso clusters de máquinas para cómputo.

Conceptualmente, CSF se adapta naturalmente a la propuesta de la RFC 2753. Los módulos del metaplanificador, colas, reserva y tareas, pueden cumplir el rol del PDP; el adaptador de recursos locales o RM Adapter cumple el rol del LPDP, y por último las máquinas virtuales cumplen el rol de PEP. Un esquema de esta propuesta puede observarse en la figura 1.

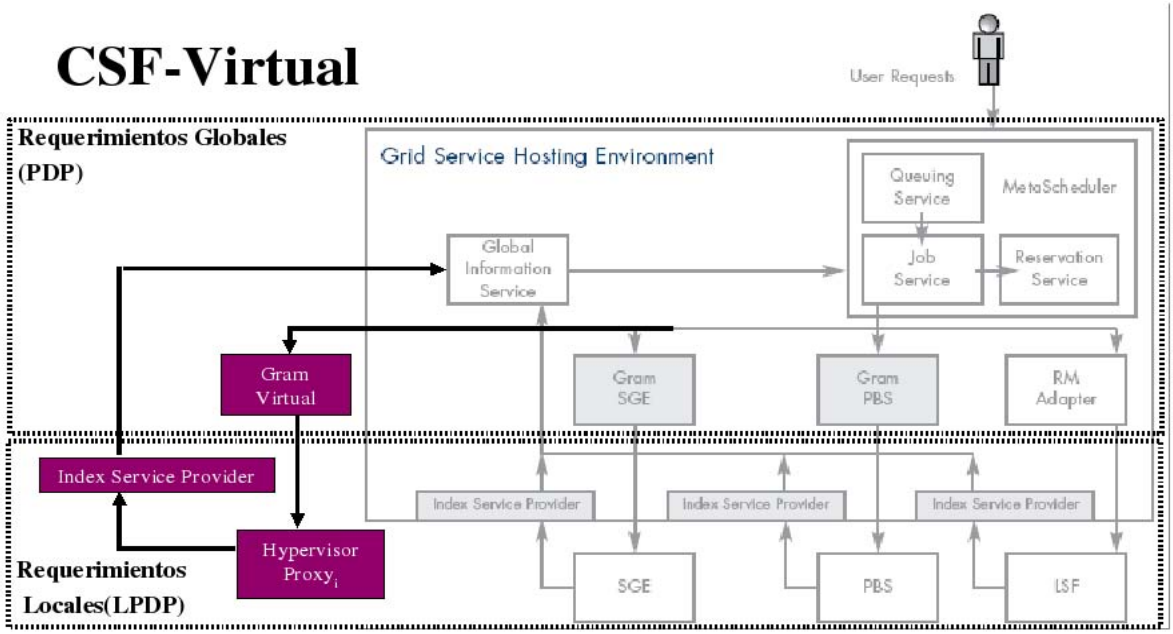


Figura 1. Arquitectura extendida de CSF

Los módulos grisados son parte del framework. Los módulos extendidos por este trabajo son las cajas rectangulares “GRAM Virtual”, este módulo es el encargado de obtener el equipamiento

solicitado por los módulos “Job Service” y “Hypervisor Proxy”, con tareas de administrador local. A diferencia de la propuesta del framework original, puede haber varios módulos “Hypervisor Proxy” coordinados trabajando en una ejecución bajo “GRAM Virtual”. Los rectangulos con bordes punteados indican la relación con la RFC antes mencionada.

Una diferencia importante que surge en la extensión y las clases provistas del framework es que los adaptadores de recursos como los planificadores locales se encuentran definidos antes de hacer el requerimiento de procesadores por parte de la aplicaciones, y si bien cada adaptador de recursos puede acceder a más de un recurso dentro de la Grid este acceso siempre es de a uno por vez. En el caso del adaptador de máquinas virtuales los recursos son dinámicos y con características heterogéneas, además por definición si la cantidad de procesadores no se logra satisfacer en un recurso físico se deberá completar el requerimiento complementándolo con otros recursos disponibles en la Grid, esto genera múltiples combinaciones y alternativas que son necesarias evaluar para obtener una solución aceptable.

Otra diferencia en la extensión del framework es el análisis estático que realizan los planificadores de cluster, una vez que las tareas fueron asignadas a sus máquinas físicas, estas se ejecutan allí sin cambios. El uso de máquinas virtuales otorga mayor libertad de gestión en tiempo de ejecución, según la magnitud de requerimientos se puede modificar la cantidad de memoria asignada, cantidad de procesadores, incluso se puede migrar la máquina virtual completa para un mejor aprovechamiento del cluster. Toda esta inteligencia debe ser incorporada en el módulo de gestión de recursos locales ya que los planificadores habituales como los que se ven en la figura 1 , SGE o PBS generalmente son solo sistemas de colas, estas características se adaptan perfectamente en el esquema de puntos de toma de decisión local.

4.1 Algoritmos de selección de máquinas

La necesidad de encontrar un conjunto de máquinas físicas con características definidas, en un tiempo razonablemente corto y que además sea una solución aceptable no es un problema trivial. Recorrer el espacio de búsqueda según alguna variable que podría ser costo o rendimiento y con todas las combinaciones de equipamiento posible no es un solución válida para la planificación, por lo que se trató de optimizar dos parámetros, el primero, la rapidez en la obtención de la solución y el segundo lo buena que puede ser la solución con respecto al caso óptimo.

Se concluyó que una alternativa aceptable es mapear la red de comunicación y las máquinas disponibles de la Grid como un grafo no dirigido, para simplificar su tratamiento. Equipos disponibles serán representados por nodos, y vínculos, por aristas. Nodos y aristas tienen un peso o costo asociado. Las aristas reciben menor peso o costo cuanto mayor ancho de banda ofrecen, y a los equipos se les asocia un peso que puede ser función del costo de alquiler del equipamiento, capacidad de procesamiento, o un mix de memoria, almacenamiento, etc. En la figura 2 se puede ver un ejemplo de un grupo de clusters. El metaplanificador realiza una consulta según los requerimientos de la aplicación y genera un grafo como el de la figura 3.

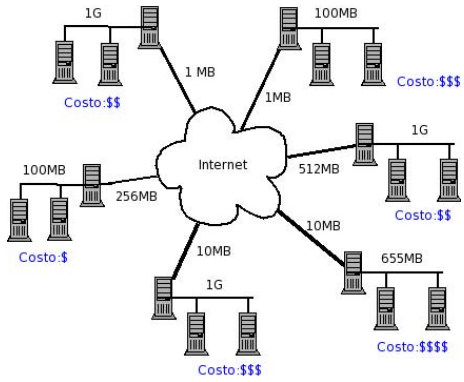


Figura 2. Ejemplo Multicluster en Grid

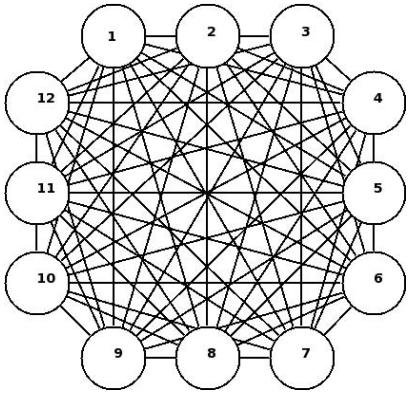


Figura 3. Mapeo de Multicluster a Grafo

Sobre el grafo se realizan búsquedas basadas en heurística como por ejemplo el algoritmo *hill-climbing*. Este algoritmo busca el camino de menor costo sin mantener un árbol de búsqueda y solo guarda el estado de su última evaluación, lo que mejora el uso de memoria, y según una función de costo determina como dar el siguiente paso. Los problemas que tiene el algoritmo son encontrar la presencia de mínimos locales y mesetas, una vez que llegue a un mínimo local a pesar de no ser una buena solución se detendrá la búsqueda, de la misma manera si encontrara mesetas, no sabría por donde continuar y podría seguir un camino equivocado. Existen maneras de mejorar el rendimiento realizando varios inicios en forma aleatoria, este método se denomina *hill-climbing con k recomienzos* donde *k* es la cantidad de nuevos inicios. Otra variante al algoritmo es no moverse siempre en el sentido correcto sino perturbando el vector de sentido de acuerdo a una función aleatoria, como es el algoritmo *simulated annealing*.

Sin embargo para este problema se desestimaron los movimientos aleatorios debido a las características singulares del problema. Puede observarse que geográficamente, agrupando países o zonas se producen conjuntos con mejor calidad o ancho de banda de comunicación que otras, esto se traduce en distanciamientos o altos costos en las aristas del grafo por conexiones con ancho de banda pobres, llevando a un particionamiento o zonificación del grafo. En estas particiones se pueden encontrar mínimos locales y alguno de estos puede coincidir con el mínimo global. Debido a este análisis se modificó la definición del algoritmo y si bien se podría seguir interpretando como k-recomienzos la idea es que estos no sean aleatorios sino que se utilicen las particiones del grafo. Esta solución no nos asegura el mínimo global pero da una buena aproximación a la misma.

Se observó que el particionamiento del grafo para el problema de los recomienzos del *hill-climbing* mejora la solución, lo cual llevó a la búsqueda de otro algoritmo para tratar el problema. Una alternativa válida es la ejecución de árboles de expansión mínimos (Minimum Spanning Trees, MST). Un árbol de expansión mínimo es un árbol al que pertenecen todos los vértices del grafo, donde la sumatoria de pesos de las aristas es menor o igual al peso de cualquier otro árbol de expansión de dicho grafo.

Existen varios algoritmos para la encontrar el MST de un grafo, el que más se adapta a la solución del problema es el algoritmo de *Kruskal*. Este algoritmo ordena las aristas del grafo de menor a mayor peso y va agregando de a una y chequeando que no se formen ciclos en el grafo hasta encontrar la solución. La selección de este algoritmo fue por su metodología de construcción del grafo, el algoritmo inicia su proceso suponiendo tantas particiones como vértices tenga el grafo. La adaptación que se realizó del problema es la suposición de que cada vértice corresponde a una máquina, por lo que si cada vez que el algoritmo agrega una arista se chequea que todas las particiones tengan al menos la cantidad de nodos que requiere la aplicación, cuando esta condición sea verdadera se tendrá la cantidad de particiones suficientes como para encontrar en forma autocontenida a cada partición una solución para ejecutar el algoritmo *hill-climbing*.

En las figuras 4 y 5 se puede observar el particionamiento según el algoritmo de *Kruskal* sobre el grafo de la figura 3. El requerimiento de la aplicación fue de cuatro procesadores por lo que se encontraron tres particiones que cumplían el requerimiento. La figura 4 muestra una solución de *hill-climbing* donde la función a optimizar fue el conjunto de máquinas más rápidas sin importar el costo. El primer paso fue buscar la máquina mas rápida por parición en cada partición; este paso está indicado en el grafo con flechas grisadas, cada uno de estos nodos sirve de inicio para el algoritmo de *hill-climbing*, de la misma manera en la figura 5 pero esta vez la función de a optimización fue el costo. En cada caso el algoritmo de *hill-climbing* realizo 3 reintentos y el máximo o mínimo local encontrado coincide con el global.

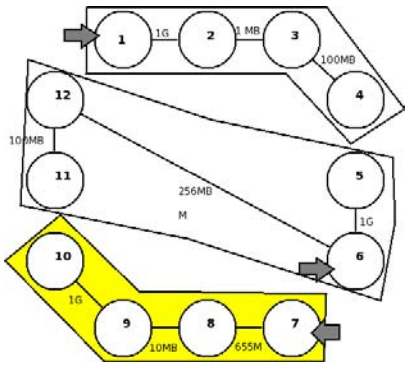


Figura 4. Solución por Heurística de Alta Prestación

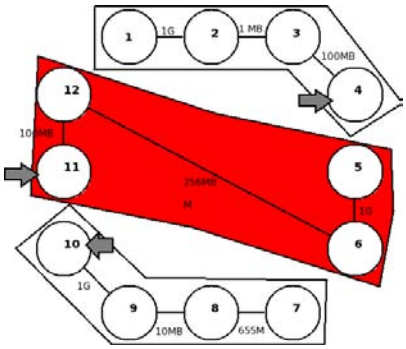


Figura 5. Solución por Heurística de Bajo Costo

Se realizó un análisis de complejidad de los algoritmos debido a la gran cantidad de aristas que se generan para un número pequeño de máquinas. Para el algoritmos MTS la complejidad es de $O(n \log(n))$ y para el algoritmo de *hill-climbing* usando lista de adyacencias es de $O(n \log(n))$. Una mejora sería detectar particiones a nivel de nodos Grid y no con todas las máquinas que integran los clusters del Grid.

5. Planificador Local

El módulo de planificación local o administrador de recursos es un servicio disponible en cada nodo Grid, posee las facultades de crear y administrar dinámicamente los recursos de la organización local. Debido a la variedad de recursos disponibles y a la complejidad de cada uno, se utilizó un protocolo de administración uniforme, este protocolo se encuentra especificado en el proyecto OASIS WSDM (Web Services Distributed Management)[8] y se utilizó la implementación Apache Muse. Este protocolo basado en servicios web permite disponer de un solo punto de acceso para un conjunto de tareas administrativas y permite la integración entre los distintos dispositivos gracias al uso de estándares.

En el caso de este trabajo los recursos involucrados en la gestión de cluster virtual son switchs virtuales, máquinas virtuales y conexiones SSH. Todos estos recursos tienen su interfaz publicada a través de WSDM. Las distintas instancias de cada recurso son registradas y accedidas desde el módulo de gestión de recursos, quien según la información obtenida del entorno ajusta dinámicamente la asignación de recursos virtuales y físicos. Para el acceso a las conexiones SSH y switches virtuales, los servicios web se implementan ejecutando comandos propios del sistema. En el caso de máquinas virtuales este aspecto es diferente, porque, por definición, la arquitectura permite el uso de distintas distribuciones de máquinas virtuales, total o parcialmente virtualizadas como por ejemplo Xen[9], KVM[10] o Qemu[11]. En este caso se utiliza la librería Libvirt para interactuar con las capacidades de cada máquina, poder interconectarlas y hacer un uso coordinado de las mismas.

Otra de las tareas, aún no implementada, del módulo de administración de recursos locales es la negociación de acuerdos de servicio con el planificador global. Cada vez que llega el requerimiento de una aplicación al planificador global, este realiza consultas a los administradores de recursos de cada organización física, cada uno de ellos según las políticas definidas en cada organización ofrecen determinados recursos con restricciones horarias o de calidad de servicios. Los módulos de administradores locales deberán hacer cumplir las restricciones impuestas por las políticas para hacer un uso racional de los recursos.

5.1 Resultados Experimentales

Para la realización de las experiencias del gestor de recursos locales se instaló un cluster con máquinas PIV de 3.06 Ghz con tecnología HT y 1GB RAM, y Intel Core 2 Duo, 1.86Ghz por núcleo. Cada máquina cuenta con sistema operativo Linux distribución CentOS 5, y kernel 2.6.18-8.1.3.el5xen. Una de ellas dispone de middleware Grid Globus 4.0.4[12], y WSDM Apache Muse 2.0.2.

Las experiencias que se realizaron tuvieron como objetivo ver la factibilidad de administración dinámica de máquinas virtuales durante la ejecución de una aplicación paralela del cluster. Se experimentó en la modificación de parámetros como memoria y cantidad de procesadores asignados a las máquinas virtuales, también con la migración de las mismas, para analizar los

efectos de estos cambios dinámicos sobre la aplicación. Para la experiencia se utilizó una aplicación de simulación con uso intensivo de CPU y baja entrada/salida y el modelo de paralelismo master-worker.

En la modificación de parámetros se utilizaron dos variables, la memoria y cantidad de procesadores. La razón de esta selección se debió a que son dos de los parámetros que generalmente se conocen a priori y son condición necesaria para la ejecución de la aplicación. En esta experiencia se sometió a la ejecución de un worker a un stress de memoria, la ejecución de la aplicación comienza con un requerimiento de memoria de 256MB y sin que la ejecución se detenga se lo disminuye. En la figura 6 se puede observar la reacción de la máquina virtual a esa modificación durante dos corridas de la aplicación. La primer corrida simbolizada por un círculo con una letra "a", es una corrida sin modificación, en la siguiente, la máquina virtual comienza con 256MB Ram y luego se le restringe a un mínimo donde debe comenzar a descargar datos a disco, esto también se refleja en el uso de CPU donde la mayor parte de los procesos son esperas de entrada/salida. Una vez que se observó este comportamiento se modificó nuevamente la memoria, ampliándola a 96MB donde la máquina retomó su normal funcionamiento. En este caso podemos ver que la máquina virtual comenzó con 256MB Ram pero solo eran necesarios 96MB para un correcto funcionamiento, encontrando de esta manera el requerimineto inferior para la ejecución.

Para las pruebas de asignación de procesadores se verificó el porcentaje de utilización fue variando proporcionalmente entre los procesos y procesadores asignados. Esto cobra importancia cuando se intenta asignar mayor prioridad a trabajos ejecutados en distintas máquinas virtuales o realizar consideraciones especiales como por ejemplo variar la cantidad de memoria por máquinas virtuales o asignar a un proceso una CPU al 100% y a otros dos a un 50% cada uno.

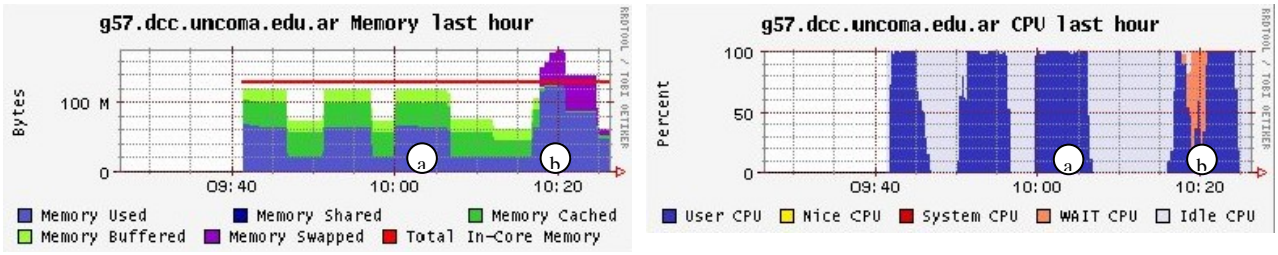


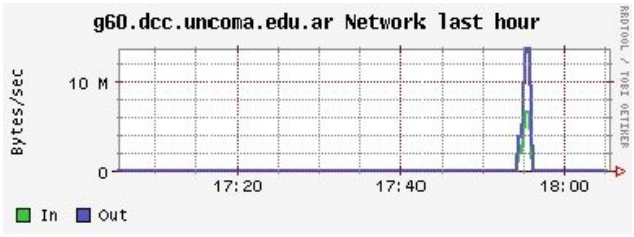
Figura 6. Modificación parámetro de memoria

Un caso donde la migración de máquinas virtuales es útil podría ser el siguiente: la configuración de un cluster es de dos equipos con doble procesador, uno dedicado a satisfacer requerimientos de la organización local y otro dedicado a los requerimientos de la organización virtual. Sin embargo los requerimientos locales tienen picos de trabajo en determinadas fechas u horas y en el resto del tiempo son esporádicos. Solo dejar dos procesadores disponibles para satisfacer los requerimientos de Grid y dejar otros ociosos sería costoso. Por lo que por política de la organización local, el administrador de recursos ofrece cuatro procesadores a la organización virtual, pero si llega algún

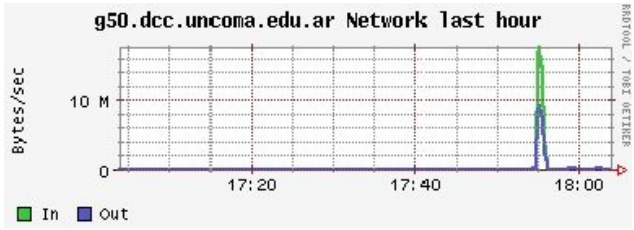
requerimiento local este tendrá prioridad sobre los procesos de la organización virtual. Si relacionamos estos conceptos con los de la figura 2, en donde cada máquina tiene un costo económico asociado, el costo será menor en el cluster en donde no necesariamente estarán todas las máquinas disponibles todo el tiempo por las políticas de la organización local.

Para el caso de uso mencionado un ejemplo sería que, en una de las máquinas se encuentra ejecutando una aplicación local a la organización, con dos procesos, cada uno utilizando su CPU asignada al 100%. Al entrar un requerimiento desde la organización virtual y como el administrador de recursos ofreció cuatro procesadores se han generado cuatro procesos sobre dos máquinas virtuales en una única máquina física, ya que la primera esta siendo ocupada por el requerimiento local, y se han asignado dos procesos a cada procesador y cada uno de ellos usa la CPU un 50%. Cuando la aplicación local termina, el administrador de recursos migra automaticamente una de las máquinas virtuales con el requerimiento global, sin detener la ejecución de la aplicación. La máquina virtual se encarga de mantener toda la información y enviarla a la nueva máquina física reasignando nuevamente los procesadores, quedando cada uno con el uso de una CPU al 100%.

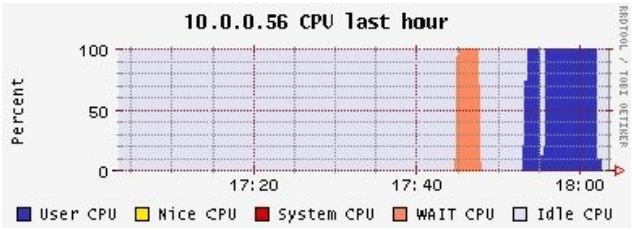
En el caso de que las políticas hubieran sido fijas una vez terminada la aplicación local, el equipamiento asignado a esas tareas hubiera quedado ocioso. De esta manera se obtiene una mejora en el aprovechamiento del equipamiento solo con adaptar las políticas.



(a) Máquina Física Origen



(b) Máquina Física Destino



(c) Máquina Virtual Migrada en Ejecución

Figura 7. Proceso de migración máquina virtual

En la figura 7 se ve un ejemplo de proceso de migración de máquinas virtuales [15]. En la subfigura (a) se encuentra el tráfico de red de entrada y salida de la máquina física origen, en la subfigura (b) el trafico de red de entrada y salida en la máquina física receptora. En la tercera subfigura de la serie se ve el proceso de simulación de la máquina virtual; como se aprecia, éste en ningún momento se interrumpe solo se atenúa unos segundos cuando migra definitivamente, no durante todo el proceso de transferencia de estado. El proceso de transferencia se hace en forma paralela a su ejecución y el impacto en la performance es mínimo.

7 Trabajo futuro

Durante el desarrollo de este trabajo se han encontrado distintas características que pueden ser explotadas para un mejor aprovechamiento del modelo propuesto. En cuanto a la selección de equipamiento, se está trabajando en incorporar múltiples criterios de selección, algunas experiencias se están orientando en poder determinar cuantas máquinas de un cluster son necesarias para lograr un rendimiento óptimo del conjunto. Hasta el momento el algoritmo busca optimizar el grafo dependiendo de una función heurística de una sola variable, pero agregando distintas variables la solución tiende a mejorar sensiblemente [13][14].

Otro tema que se encuentra en estudio es la migración interdominio, el ejemplo propuesto en este trabajo migra máquinas virtuales solo en un cluster y esto reduce de manera considerable el uso de la red. Las imágenes de las maquinas virtuales se encuentran compartidas a través de un repositorio común con NSF, una de las alternativas que se barajan es usar la característica *copy-on-write* para minimizar la transferencia.

El último punto en el que se está trabajando es la parada de la aplicación distribuida en el entorno Grid. Esto tiene relación con conceptos de sistemas distribuidos de estado global del sistema, se está analizando colocar monitores en las colas de mensajes de los hypervisores y detectar cuando hay un estado estable del sistema para un dominio restringido de aplicaciones.

8 Conclusión

En este trabajo se han presentado la extensión de un metaplanificador para la generación de laboratorios remotos de máquinas virtuales y administradores de recursos locales y su posterior administración; dentro de un esquema de administración basado en políticas.

Se han implementado algoritmos para la búsqueda de un conjunto de máquinas que satisfaga el requerimiento del uso de un espacio reducido de búsqueda, tratando de minimizar el tiempo de procesamiento y la calidad de la solución.

En el administrador local de recursos se han verificado los beneficios del uso de máquinas virtuales para lograr un máximo aprovechamiento de los clusters pudiendo modificar de manera homogénea parámetros de ejecución como memoria y cantidad de procesadores en forma dinámica y migrar máquinas virtuales reasignando el espacio de máquinas físicas sin tener que detener la aplicación que está corriendo en el cluster.

Estos resultados permitieron concluir que ésta tecnología es factible de aplicase en espacios Grid donde existe la necesidad de adaptación y control del entorno para requerimientos de diversa índole. En éste sentido redunda en beneficios importantes para la calidad de administración y una mejor utilización de recursos.

9 Referencias

[1] I. Foster, T. Freeman, K. Keahey, D. Scheftner, B. Sotomayor, X. Zhang. “Virtual Clusters for Grid Communities”. CCGrid 2006.

[2] Chase, J., L. Grit, D. Irwin, J. Moore, and S. Sprenkle, “Dynamic Virtual Clusters in a Grid Site Manager”. accepted to the 12th International Symposium on High Performance Distributed Computing (HPDC-12), 2003.

[3] Paul Ruth, Phil McGachey, Dongyan Xu, "VioCluster: Virtualization for Dynamic Computational Domains", Proceedings of the IEEE International Conference on Cluster Computing (Cluster'05), 2005.

[4] Sumalatha Adabala, Vineet Chadha, Puneet Chawla, Renato Figueiredo, Jose A. B. Fortes, Ivan Krsul, Andrea Matsunaga, Mauricio Tsugawa, Jian Zhang, Ming Zhao,Liping Zhu, Xiaomin Zhu, “From Virtualized Resources to Virtual Computing Grids: The In-VIGO System”, In Future Generation Computing Systems, 2004.

[5] Eduardo Grosclaude, Francisco López Luro, Mario Leandro Bertogna,”Grid Virtual Laboratory Architecture”. VHPC Euro-Par'07. 2007.

[6] Yavatkar, R., Pendarakis, D. and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000.

[7] Platform Computing Co. Open source metascheduling for Virtual Organizations with the Community Scheduler Framework (CSF)[WP]. www.cs.virginia.edu/~grimshaw/CS851-2004/Platform/CSF_architecture.pdf , 2004.

[8] OASIS “An Introduction to WSDM”, February,2006, http://www.oasisopen.org/committees/download.php/16998/wsdm-1.0-intro-primer-cd-01.doc.

[9] Paul T. Barham, Boris Dragovic, Keir Fraser, Steven Hand, Timothy L. Harris, Alex Ho, Rolf Neugebauer “Xen and the Art of Virtualization”, SOSP 2003 , Pages 164-177.

[10] KVM: “Kernel Based Virtual Machine”. http://kvm.qumranet.com/kvmwiki

[11] Fabrice Bellard. "QEMU, a fast and portable dynamic translator", Proceedings of USENIX. April 2005.

[12] I. Foster, C. Kesselman, S. Tuecke.”The Anatomy of the Grid: Enabling Scalable Virtual Organizations”. International J. Supercomputer Applications, 15(3), 2001.

[13] Viktor Yarmolenko, Rizos Sakellariou. “An Eevaluation of Heuristics for SLABased Parallell Job Scheduling”**.**Proceedings to the Third High-Performance Grid Computing Workshop(HPGC), IEEE International Parallel & Distributed Processing Symposium (IPDPS'06) in Rhodes Island, Greece. 2006

[14] Eduardo Argollo, Adriana Gaudiani, Dolores Rexachs, Emilio Luque.”Tuning Application in a Multi-cluster Environment”. Euro-Par 2006 . 78-88, 2006.

[15] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen.”Live Migration of Virtual Machines”. Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation. 2005.

Problemas de la Sincronización de Relojes en Clusters

Fernando L. Romero, Fernando G. Tinetti*

III-LIDI, Facultad de Informática, UNLP
50 y 115, 1900, La Plata, Argentina
fromero@lidi.info.unlp.edu.ar, fernando@info.unlp.edu.ar

Resumen

Se presenta un trabajo orientado a resolver algunos aspectos del problema de sincronización de relojes en ambientes distribuidos. El objetivo inicial está dirigido a la estimación de rendimiento en estos ambientes, donde es necesario algún tipo de sincronización de relojes de las computadoras que se utilizan. El algoritmo básico que se utiliza es una modificación del algoritmo de Cristian, considerando necesario adaptarlo al entorno de un cluster o, al menos, de una red de interconexión sobre la que se tiene información y acceso relativamente exclusivo o previsible para todas las comunicaciones entre las computadoras que se sincronizan. Una de las ideas subyacentes es la de aislar la sincronización de la ejecución de la aplicación cuyo rendimiento se quiere evaluar. La finalidad de este requerimiento es evitar una intervención de la instrumentación sobre los valores a medir, o al menos saber qué error tendrán los mismos. El ambiente en el cual se realizan los experimentos es inicialmente el de los entornos de cómputo paralelo en clusters. Se ensayan diferentes métodos para la sincronización en valor inicial y frecuencia de los relojes.

Palabras claves: *Sincronización de Procesos, Relojes Distribuidos, Rendimiento e Instrumentación, Sistemas Paralelos y Distribuidos, Paralelismo en Clusters e Intercluster, Sincronización Interna y Externa .*

Abstract

This paper is oriented to solve some of the aspects of the clock synchronization problem in distributed environments. Its main objective is focused on the performance estimate in these environments, in which it is necessary to count with some kind of clock synchronization of the computers used. The basic algorithm used is a modification of the Christian's algorithm, which must be adapted to a cluster's environment or, at least, to an interconnection network about which some information is available and to which there exist a relatively exclusive or predictable access for all the communications among the synchronized computers. One of the underlying ideas is that of isolating the application running synchronization, whose performance is to be evaluated. The end sought with this requirement is to avoid the intervention of instrumentation over the values to be measured or, at least, to know which errors they might present. The environment in which experiments are carried out is initially that of cluster parallel computing environments. Different methods are tested for the synchronization in initial values and clock frequency.

Keywords: *Process Synchronization, Distributed Clocks, Performance and Instrumentation, Parallel and Distributed Systems, Parallelism in Clusters and Interclusters, Internal and External Synchronization.*

* Investigador Asistente CICPBA

1 INTRODUCCION

Las computadoras poseen uno o varios relojes que proporcionan una referencia de tiempo [16] [22]. Dicha referencia es esencial para resolver problemas tales como el ordenamiento de eventos (ej: envío y recepción de correo electrónico, eventos dentro de las transacciones, inicio de procesos en tiempo real, etc.) como para tareas propias del sistema, tales como planificación de procesos. Estos relojes permiten la medición de intervalos de tiempo, necesarios para la optimización del rendimiento tanto en sistemas monoprocesador como en sistemas paralelos y distribuidos [15] [4] [10]. En todos los casos, las aplicaciones con fuertes requerimientos de cómputo, procesamiento y comunicaciones son las que también requieren la optimización para el máximo aprovechamiento del hardware disponible. A partir de la monitorización de los tiempos de ejecución se pueden analizar los problemas de rendimiento e intentar solucionarlos [6].

Al realizar mediciones de tiempo de ejecución de procesos y de comunicaciones es deseable que el registro de tiempos no influya en el tiempo de ejecución de los mismos. Debido al incremento de las velocidades de los procesadores, la disminución de las latencias en las comunicaciones y el aumento del ancho de banda en las mismas, dado por la evolución del hardware, el requerimento de resolución en las medidas de tiempo fue creciendo, estando en la actualidad en el orden de los microsegundos. Los métodos provistos por el sistema operativo no son apropiados [17] debido no solo a problemas de resolución sino también por el nivel de interferencia (Ej: llamadas al SO). Los métodos y/o herramientas provistas por los lenguajes heredan este problema ya que dependen del sistema operativo. La solución es una mejor uso de las nuevas características del hardware. Ejemplo es el uso de la instrucción RDTSC (ReaD Time Stamp Counter) [16] [22].

En el caso de procesamiento distribuido, con un programa que ejecuta procesos en diferentes computadoras o en los que el tiempo de las comunicaciones es importante, la tarea de medir intervalos de tiempo implica sincronizar los relojes de las diferentes computadoras que se utilizan [9]. Sería deseable que esta tarea de sincronización se lleve a cabo fuera del tiempo en que se ejecute el programa que se está monitorizando, y conociendo el tipo (o al menos magnitud) de error con que se sincroniza. Es requerimiento de este trabajo que dicha sincronización se lleve a cabo sin la necesidad de incluir hardware adicional al del sistema, con lo que las comunicaciones deberán utilizar la red de interconexión entre computadoras. Asimismo, como sistema de medición, aprovechar el hardware de cada sistema de cómputo, sin agregados ni cambios.

En principio, esta herramienta de instrumentación sería utilizada para evaluar programas paralelos, cumpliendo que:

- Pueda ser usada inicialmente en un cluster de PC's, con la posibilidad de ser extendido a clusters en general y luego en plataformas distribuidas aún más generales.
- Sea de alta resolución, es decir que se pueda utilizar para medir tiempos cortos, del orden de microsegundos.
- Que no altere el funcionamiento de la aplicación bajo prueba, o que la alteración sea mínima y conocida por la aplicación.
- Utilice en forma predecible la red de interconexión. Más específicamente, que se puedan determinar, desde la aplicación, los intervalos de tiempo en los cuales se utilizará la red. De esta forma, se puede *desacoplar* el uso de la red de interconexión, ya que habrá intervalos de tiempo usados para la sincronización e intervalos de tiempo utilizados para la ejecución de programas paralelos.

2 TRABAJOS PREVIOS RELACIONADOS

Se estudiaron tanto los algoritmos básicos como las implementaciones existentes. Como requisito previo, se establece que cada computadora cuente con un reloj físico de frecuencia más o menos constante. En general, estos relojes utilizan como estabilizador del oscilador un cristal de cuarzo. Presentan valores de variabilidad de frecuencia que van desde 10^{-4} a 10^{-6} [18]. A partir de este reloj físico se derivan los relojes lógicos que son los que se sincronizan [5] [8] [13]. En todos los casos, lo que se tiende a resolver son las diferencias de [19] [20]:

1. Referencia fija en el pasado a partir de la cual se contabiliza el tiempo en cada computadora.
2. Frecuencia entre los relojes de las computadoras que se sincronizan.

Una vez realizada la sincronización mínima entre las computadoras se debe investigar el comportamiento de la misma en términos de escalabilidad. Usualmente la sincronización se da entre dos máquinas. La propuesta inicial de Cristian [3], de contar con un servidor que proporcionara una hora ajustada a un determinado estándar (sincronización externa) y clientes sincronizando con este servidor, sirvió de base para el desarrollo de protocolos más complejos en los que se consideraron otros aspectos del problema. Podemos citar a NTP (Network Time Protocol) y el algoritmo de Berkeley como ejemplos, siendo este último un ejemplo de sincronización interna. En cuanto a ordenación de eventos sin un estándar en la hora, el algoritmo de Lamport es un ejemplo de otra orientación para solucionar el problema de sincronización. Es claro que cualquier tipo de centralización (en el servidor, por ejemplo) tiene sus inconvenientes de escalabilidad y al menos debería ser posible su cuantificación. Referido a análisis y experimentación de sistemas existentes, se han llevado a cabo pruebas con NTP y con el Algoritmo de Berkeley específicamente con su implementación en Linux, el timed (**time d**aemon). También se analiza a continuación el algoritmo de Lamport.

2.1 NTP: Network Time Protocol

Una de las primeras consecuencias de utilizar NTP está relacionada con la precisión del reloj obtenido, independientemente de la sincronización. En el caso de la instalación de NTP realizada en computadoras con Linux 2.4.18-14, sobre PCs con procesador Intel Pentium 4 de 2.4 GHz y 1 GB de RAM, la precisión del reloj de NTP es de 2e-17s = 7,6 μs. En estas mismas computadoras, la resolución del reloj sin NTP es de 1 μs, con lo que en cierta forma se pierde algo de precisión *a priori*. NTP corrige dos tipos de diferencias (*offsets*) para sincronizar dos o más computadoras:

- ***Offset* de hora:** Va realizando ajustes de acuerdo a que la diferencia respecto de la referencia sea grande o pequeña. Si la diferencia es mayor a 128 μs. Se da en un escalón (*stepping*) y si es menor a 128ms, en forma gradual (*slewing*). En todos los casos, si el reloj local esta adelantado, va haciendo ajustes graduales de tal manera de no producir discontinuidades en la hora (hora posterior < hora actual).
- ***Offset* de frecuencia:** a partir de analizar la primera derivada de la diferencia de horas, establece si la frecuencia es mayor o menor a la de la referencia, cambiando el valor local de frecuencia al correcto.

En la Tabla 1 se pueden observar algunos valores relacionados con la sincronización de una máquina con respecto a un servidor en la misma red local. Las computadoras son iguales, con las características ya mencionadas, y la red de interconexión es Ethernet de 100 Mb/s dedicada (no hay otro tráfico en la red) con *switch* de capa 2. Mientras no posee suficientes datos como para asegurar sincronización, NTP informa que no está sincronizada, y el valor de hora dependerá del reloj local. A partir de alcanzar una sincronización mínima a los 6 minutos de haber comenzado, NTP ya proporciona los valores corregidos. A partir de este primer instante de sincronización, los valores

relacionados mejoran considerablemente, aún después de varias horas de funcionamiento de NTP.

Tabla 1: Valores de Sincronización de NTP.

Hora	*Offset*(s)	*Dist. Sync.(s)*	*synch?*
15:46:14	0,000018	0,00375	Not
15:47:00	0,000019	0,00444	
15:48:00	0,000022	0,00534	
15:49:00	0,000016	0,45090	
15:50:00	0,000016	0,45180	
15:51:00	0,000018	0,45271	
15:52:00	-0,000369	0,01141	Yes
15:53:00	-0,000431	0,01134	
15:54:00	-0,000442	0,01126	
15:55:00	-0,000446	0,01120	
15:56:00	-0,000451	0,01111	
15:57:00	-0,000459	0,01105	
15:58:00	-0,000457	0,01097	
15:59:01	-0,000461	0,01187	
16:00:00	-0,000455	0,01178	

El valor que proporciona NTP para estimar la magnitud del error de sincronización en un instante dado es la distancia de sincronización (Dist.Sync. en la Tabla 1). Este parámetro se calcula en base a una estadística en los tiempos de comunicación con el servidor de referencia, la distancia de sincronización con que el servidor a su vez recibe los valores de referencia, y los valores registrados de *offset* de frecuencia. Las variaciones de frecuencia (o posible derivada segunda de la hora) se registran e incorporan al cálculo de la distancia de sincronización. El pseudocódigo del cálculo de dicho parámetro puede verse en el RFC-1305.

La dificultad de contar con la distancia de sincronización como única estimación de error, es que no se puede saber con certeza el error de sincronización sino una cota, que normalmente es del orden de los milisegundos. En la Tabla 1 se puede verificar que se sincroniza a valores de alrededor de 400 μs (*offset*) con un posible error acotado de poco más de 1100 μs, valores alejados de los requeridos en este trabajo. Con respecto a la sobrecarga en comunicaciones, se puede configurar la frecuencia a la que se interrogará al servidor. En caso de querer eliminar toda intrusión de NTP en las medidas de rendimiento que se deban realizar, sería necesario detener los procesos de NTP, con lo que los valores de error aumentarán, y en forma imprevisible, pues el reloj empezará a tomar como referencia el reloj local, con las correcciones hechas por NTP en el período en que estuvo funcionando. De otra manera, habría que admitir el "ruido" de los paquetes de sincronización, normalmente en sucesiones (o *ráfagas*) de 4 paquetes UDP, que también implican el costo de procesamiento por parte de los de NTP, con las posibles llamadas al sistema para ajustar los valores de la hora.

2.2 Algoritmo de Berkeley

A diferencia de NTP, el algoritmo de Berkeley tiene un proceso (o computadora) *Master* que se encarga de definir un tiempo sincronizado a partir del tiempo local de todos los demás procesos (o computadoras). De manera periódica, el Master

1. Requiere la hora local de cada uno de los demás procesos.
2. Calcula el promedio de los valores recibidos (descartando los que difieran demasiado).

3. Informa a los demás procesos el cambio que deben realizar para estar sincronizado.

En el cálculo de cada nuevo valor de tiempo a utilizar (con el promedio de los tiempos recibidos) no se tiene en cuenta el tiempo de comunicaciones necesario para las transferencias de datos hacia y desde el proceso *Master*.

Este algoritmo fue uno de los primeros incluidos en las distribuciones más conocidas de Linux, aunque algunas de ellas ya no lo incluyen más. Siempre es posible (y sencillo, de hecho) instalarlo en Linux, donde básicamente se pone en marcha el proceso timed a nivel de sistema operativo. En todos los casos la mejor resolución obtenida es de 1 milisegundo, a la que se llega bastante más rápidamente que con NTP. En principio, no cumple con uno de los objetivos originales de este trabajo que es llegar al orden de los microsegundos.

2.3 Algoritmo de Lamport

El algoritmo de Lamport es uno de los primeros propuestos para la sincronización de sistemas distribuidos [2] y se basa en la relación "sucede antes" más la utilización de los mensajes entre las computadoras como indicadores precisos de esta relación. Más específicamente, un mensaje no puede ser recibido antes de ser enviado y, por lo tanto, si se tienen marcas de tiempo de los envíos de los mensajes se puede verificar si el tiempo actual es coherente con la definición de la relación "antes de". Las tareas a llevar a cabo en cada computadora son relativamente sencillas, aunque afectan, en cierta manera, la forma en que se procesan los mensajes:

1. Se tiene un reloj local.
2. Cada vez que se envía un mensaje, se le *agrega* al mismo una marca de tiempo (*timestamp*) con el tiempo local del que envía.
3. Cada vez que llega un mensaje, se analiza la marca de tiempo del que envió, y
    a. Si la marca de tiempo es menor que el tiempo local, se asume que las computadoras están sincronizadas
    b. Si la marca de tiempo es mayor que el tiempo local, se cambia el tiempo local con la marca de tiempo del mensaje que se recibió más 1 (asumiendo, por ejemplo, que la transmisión necesita 1 unidad de tiempo)

La Fig. 1 muestra el avance del tiempo en dos computadoras, C1 y C2, y la forma en que la computadora C2 cambia su tiempo local a partir de la llegada de un mensaje con una marca de tiempo mayor que su tiempo local.

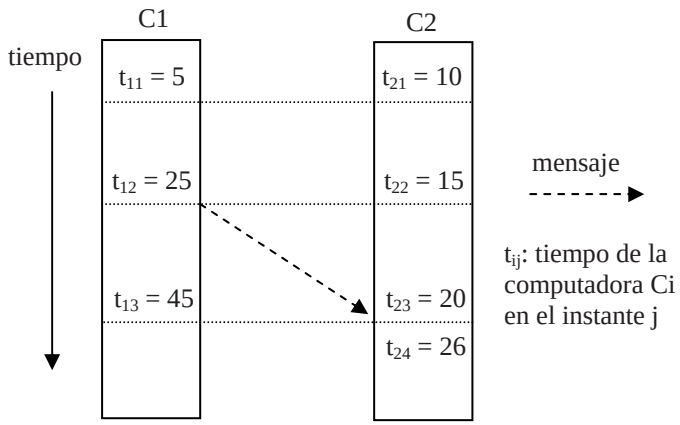


Figura 1: Ejemplo de Funcionamiento del Algoritmo de Lamport.

Este algoritmo soluciona el problema de la escalabilidad, pero no tiene un orden total de los eventos que suceden entre diferentes computadoras excepto los relacionados directamente con envío y recepción de mensajes. En el caso de la Fig. .1, no es posible determinar la relación de tiempos entre los eventos que suceden en la computadora C1 entre los instantes de tiempo t_{12} y t_{13} con los que suceden en la computadora C2 entre los instantes de tiempo t_{22} y t_{23}.

3 ANALISIS DE DIFERENCIAS ENTRE RELOJES

Los valores de hora en dos computadoras pueden diferir a partir de un instante de tiempo por un cierto error inicial (*offset*), e irá creciendo en el tiempo dado por las diferencias en sus frecuencias (primera derivada) más un error aleatorio dado por la deriva en el valor de frecuencia de cada una de las frecuencias, producto de cambios ambientales [15] [14]. Al menos la diferencia de dos relojes en un instante de tiempo puede ser estimado en una computadora a partir del tiempo de transmisión entre las dos máquinas. Si dicho tiempo fuese constante, se podria medir y eliminar este error utilizando la constante necesaria. Como ambos relojes no están sincronizados, el timepo debe ser medido por un solo reloj, por lo tanto se mide el tiempo de ida y vuelta de un paquete en la red de comunicaciones (*round trip time*). Se estima el tiempo de transmisión como la mitad del tiempo de ida y vuelta, lo cual no necesariamente es correcto. Una descomposición de este tiempo sería el de la Fig. 2, donde

- **Tiempo de envío**: Es el gastado en la construcción del paquete en las distintas capas hasta alcanzar la capa de acceso al medio (*MAC*). Es variable en función del estado del SO (cambio de contexto, planificación, etc.).
- **Tiempo de acceso al medio**: Depende del protocolo usado. En el protocolo Ethernet dependerá del tráfico en la red. Es aleatorio.
- **Tiempo de transmisión**: Depende de la velocidad de la placa de red y del largo del mensaje. Es determinístico.
- **Tiempo de propagación**: en una red local será despreciable ya que es lo que tarda en recorrer el cable entre placas de red - *switch* - placa de red más el retardo que introduce el *switch*.
- **Tiempo de recepción**: Es el tiempo que tarda en atravesar hasta la *MAC layer*. Es determinístico.
- **Tiempo de recibo**: Es el gastado en desarmar el paquete en las distintas capas hasta ser entregado a la aplicación. Depende del SO como el de envío.

Figura 2: Descomposición del Tiempo de un Mensaje entre Dos Computadoras.

Estos tiempos son difíciles de medir sin contar con equipamiento especializado de mediciones electrónicas. El principal problema que introducen referido a la sincronización de relojes es debido a que son variables, con lo que sólo se los puede tratar estadísticamente o directamente acotar. Otro enfoque es trabajar sobre los elementos de *hardware* y *software* (principalmente sistema operativo) que causan que sean variables. Sería deseable que aunque no se pueda eliminar el error, se pueda minimizar y además tener una estimación del mismo. Para ello se realizaron experimentos tendientes a medir la moda, valor mínimo y máximo en un ambiente como el que se especifica en los requerimientos. Para acotar aun más la parte variable, se midió el mínimo tiempo que tarda el sistema en acceder a los *buffers* de comunicación. A partir de estos experimentos, se determina que todas las trasmisiones se tomen como válidas solamente si fueron realizadas en un tiempo rtt

(*round-trip time*, tiempo de ida y vuelta de un paquete) igual a la moda. Con ello se tiende a acotar los errores debido a las comunicaciones tanto en la determinación del *offset* inicial entre relojes como de la diferencia de frecuencia. Un problema agregado es el de la asimetría entre los tiempos de ida y vuelta. Si se toman como válidas solo las referencias de tiempo trasmitidas en tiempo rtt igual al valor de la moda, es muy probable que dicha asimetría puede estar dada por la diferencia entre la moda y el mínimo, actualmente se está estudiando si es posible evitar el error por asumir que los tiempos de ida y vuelta de un mensaje son iguales.

Una vez resuelto el problema del *offset inicial*, se deben tener en cuenta las diferencias de frecuencias con las cuales se actualiza la hora en cada computadora. Los sistemas operativos como Linux usualmente tienen calculado un valor de frecuencia del oscilador a partir del cual actualiza la hora del sistema. Este valor, generalmente dado en MHz (millones de ciclos por segundo), no es necesariamente correcto para el registro de la hora con precisión de microsegundos. Por esta razón se considera adecuado hacer el cálculo de MHz del oscilador de cada computadora de manera tal que el cálculo de la hora posterior utilizando este valor tenga precisión del orden de microsegundos o mejor. Como se aclaró en el caso de NTP, tener un reloj local (o los relojes locales de todas las computadoras) con precisión del orden de microsegundos no necesariamente implica que el error de sincronización sea de esta magnitud.

El cálculo de MHz reales de una computadora es relativamente sencillo a partir de RDTSC y el uso del reloj de *hardware* del sistema. Aunque la precisión del reloj del sistema no sea muy apropiada para el registro de tiempo del orden de microsegundos, la frecuencia del mismo es suficientemente estable como para ser considerada constante en un intervalo de tiempo suficiente como para el cómputo inicial de MHz. Asumiendo que el oscilador con al que se accede con RDTSC es de frecuencia constante, el cálculo de MHz puede ser realizado en un único punto en el tiempo, con lo cual este tiempo inicial puede ser relativamente grande (segundos o quizás minutos). Sin embargo, el cálculo no necesariamente es tan sencillo en diferentes computadoras que luego se deben sincronizar. La razón fundamental está relacionada con el reloj de hardware de referencia con el cual se realiza el cálculo de MHz de cada computadora.

La alternativa es, justamente, que una sola de las computadoras, usualmente la que funciona como servidor de hora, proporcione las referencias de hora con la cual se hace el cálculo de MHz en cada una de las demás computadoras. Este esquema es sencillo y también se pueden evitar errores haciendo mayor el tiempo durante el cual se contabilizan las oscilaciones en base a las cuales se lleva a cabo el cálculo de MHz. En cierta forma, tanto las referencias de tiempo del reloj de hardware de una PC como las referencias de tiempo que se reciben desde otra computadora tienen un margen de error, que se puede disminuir proporcionalmente con el propio tiempo que se contabiliza. En este caso también se utiliza la red de interconexión y se toman los tiempos de mensajes como significativos para el control del error en el cálculo de MHz. Más específicamente:

- En las computadoras utilizadas de 2.4 GHz se tienen aproximadamente 2.4×10^9 oscilaciones por segundo.
- Un paquete TCP utiliza aproximadamente 120 μs como tiempo de ida y vuelta, con lo cual se puede asumir que se necesitan aproximadamente 60 μs para un envío-recepción de un paquete TCP individual (o *one-way transmission time*).
- Durante un tiempo de 10 segundos (10^7 μs) en el servidor, el tiempo que se contabiliza en los clientes a partir de los mensajes recibidos podría tener el error de tiempo de transmisión dos mensajes, es decir 120 μs, con lo cual el posible error está acotado por $120/10^7$, es decir que el error en la contabilización de oscilaciones es de $120/10^7$, o de otra manera, si k es la cantidad de oscilaciones contabilizadas, el valor real puede variar $\pm120/10^7\times k$.

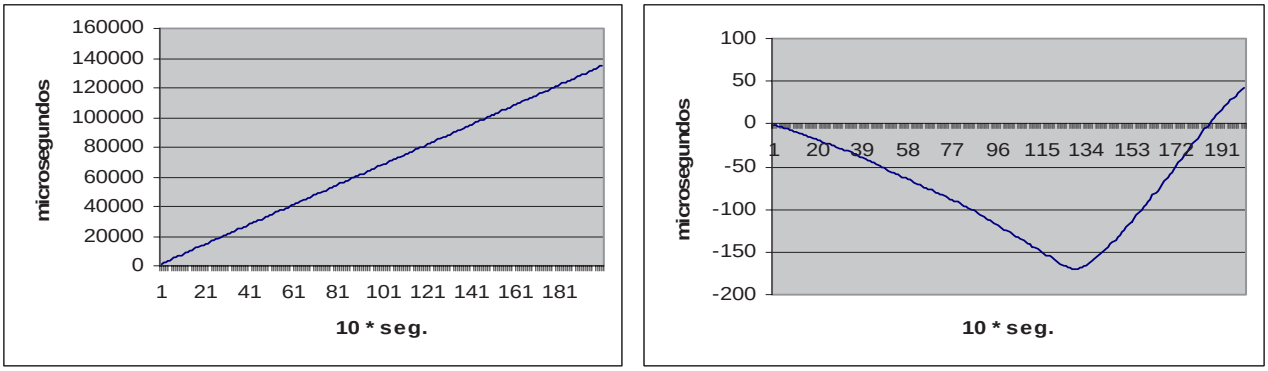
La Tabla 2 muestra los valores que se obtienen al sincronizar relojes con cálculo de MHz totalmente independientes (con los relojes locales) y con cálculo de MHz con referencia única, es decir donde una sola computadora proporciona las referencias de tiempo para el cálculo de MHz. El experimento se llevó a cabo sincronizando desde 2 hasta 17 máquinas, y luego se midió la diferencia 1 segundo después. En todos los casos son valores mínimos y máximos luego de 50 corridas. La Tabla 2 muestra los errores mínimo y máximo de sincronización y debe recordarse que el error de sincronización se define como la diferencia en los tiempos locales de las computadoras y estos tiempos, a su vez, tienen una resolución de aproximadamente 1.18 μs [20].

Tabla 2: Errores de Sincronización por Cálculo de MHz.

Cant. Máq.	**Ref.única**		**Ref.local**	
	Mín.	Máx.	Mín.	Máx.
2	-1	3	42	103
3	-3	3	56	116
4	-3	3	55	98
5	-2	3	55	114
6	-2	3	57	244
7	-3	3	56	1533
8	-3	4	56	325
9	-3	3	56	210
10	-3	4	-4	1308
11	-4	4	41	1395
12	-4	5	58	1802
13	-4	5	57	1277
14	-3	3	43	3808
15	-3	4	43	1853
16	-4	3	43	1109
17	-14	6	50	2128

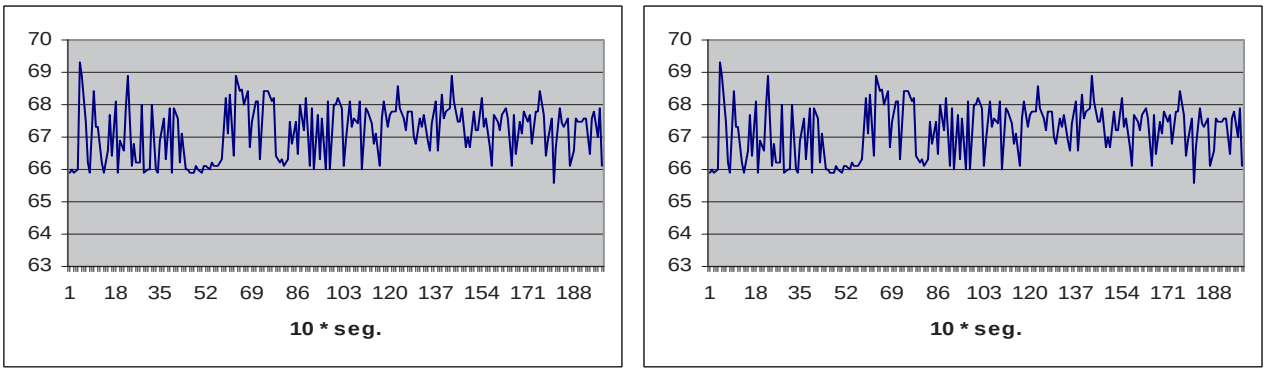
En la Fig. 3 se puede ver la diferencia de hora a partir de un ajuste inicial entre dos máquinas cuya frecuencia de reloj ha sido corregida tomando referencia local y referencia única. Se debe notar la diferencia en cuanto a los valores de las diferencias: para la Fig. 2a) varía entre 0 y poco menos de 150000 μs, y para la Fig. 2b) varía entre aproximadamente -150 y 50 μs. Obsérvese que con referencia local, Fig. 2a), la diferencia aumenta proporcionalmente a medida que transcurre el tiempo y en aproximadamente media hora sin resincronizar, se ha apartado del valor de la hora de la máquina de referencia alrededor de 0,14 segundos. En el caso de referencia única, Fig. 2b) al transcurrir el mismo tiempo la diferencia máxima fue 0,000150 segundos (150 μs.). Es de hacer notar que en vez de aumentar constantemente, vuelve a converger al valor de la hora de referencia. Estas variaciones corresponden a variaciones de tipo ambiental, como podrá verificarse en la primera y segunda derivada. La Fig. 4 muestra los valores relacionados con la primera derivada de los valores de hora. En el caso de referencia local, Fig. 4a), este valor es relativamente grande y siempre positivo, lo cual implica el constante aumento de las diferencias de tiempo. En el caso de referencia única, el cambio en el signo hace converger las diferencias de hora hacia valores menores. Como dato adicional, se grafican en la Fig. 5 los valores de la segunda derivada, donde se observa que tanto para referencia local como única son similares (0 a 0,3). Esta derivada es la que

muestra los cambios debidos a factores ambientales, que normalmente son difíciles de controlar más alla de recomendaciones tales como tener las máquinas en el mismo cuarto o en condiciones climáticas similares, con las medidas de refrigeración acordes al hardware, etc.



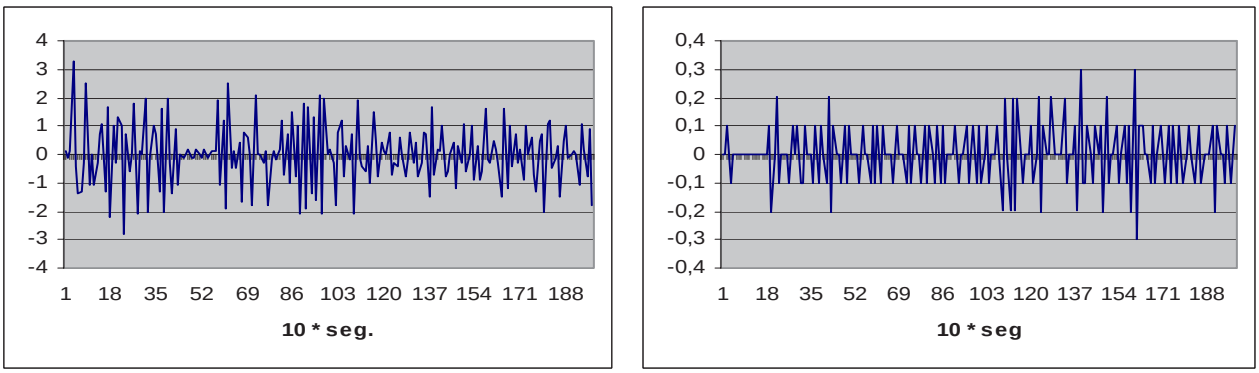
a) Referencia Local b) Referencia Unica.

Figura 3: Diferencias de Tiempo Dependiendo del Cálculo de MHz.



a) Referencia Local b) Referencia Unica.

Figura 4: Primera Derivada de las Diferencias de Tiempo Dependiendo del Cálculo de MHz.



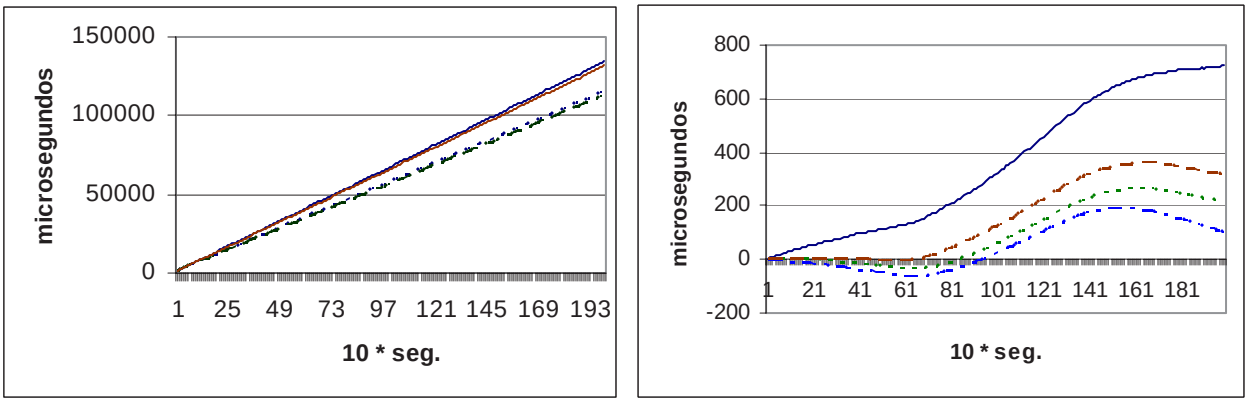
a) Referencia Local b) Referencia Unica.

Figura 5: Segunda Derivada de las Diferencias de Tiempo Dependiendo del Cálculo de MHz.

En todos los experimentos se tienen en cuenta las diferencias entre relojes, ya que para ver solamente las diferencias del cliente, el servidor debería proporcionar un reloj perfecto. Una posible mejora sería disponer de una máquina con sistema operativo de tiempo real para suministrar la

referencia de tiempo. Otra forma adicional de mejorar los valores a medir sería tener un reloj de referencia mejor. Como la idea es no recurrir a ningun tipo de hardware adicional, una de las posibilidades es armar un ensamble de relojes. Se demuestra que un arreglo de relojes puede dar una hora más estable [21][1] y también puede tomar la idea del algoritmo de Berkeley [2].

En la Fig. 6 se pueden ver las diferencias de hora de 4 clientes con respecto al servidor. En el caso de referencia local, Fig. 6a), luego de 2000 segundos sin resincronizar el desfasaje máximo supera los 30000 microsegundos. Con referencia única, Fig. 6b) dicho valor es de 750 microsegundos, o sea unas 40 veces más pequeño. Esto demuestra que, más allá del error inicial que puede ser aproximadamente igual en ambos casos, se obtiene un error en frecuencia sensiblemente menor, lo que llevará a una menor pérdida de sincronización a medida que transcurre el tiempo, evitando una resincronización de relojes para mediciones de tiempos relativamente largos. El hecho que un grafico presenta rectas y el otro curvas está relacionado con la escala del eje y, en realidad la recta no es tal, sino su derivada segunda sería 0.



a) Referencia Local b) Referencia Unica.

Figura 6:Diferencias de Tiempo de 4 Clientes con Respecto al Servidor de Hora.

4 CONCLUSIONES

La utilización de una única referencia de tiempo para la corrección del error de frecuencia en las diferentes máquinas en lugar del propio reloj local presenta una reducción importante de los errores remanentes. El algoritmo planteado, una modificación al algoritmo de Cristian [3] [5], ha demostrado su eficacia en las pruebas en un cluster. En un ambiente fuera de red local los errores pueden ser demasiado grandes respecto de las mediciones que se deban realizar debido a la variabilidad de los tiempos de comunicación.

Con respecto a la escalabilidad, el modelo cliente servidor utilizado puede presentar problemas, una posible solución sería probar con comunicaciones *broadcast* para la sincronización. De hecho, se deberían estimar/evaluar los problemas de escalabilidad y en función de los valores observados proponer una posible solución. También sería necesario evaluar la sobrecarga que se impone en el servidor específicamente con respecto a la cantidad de clientes, algo que también está relacionado con la escalabilidad del esquema de sincronización planteado en este trabajo.

Como extensiones futuras, siempre es deseable la sincronización externa de los relojes [5]. Este paso está muy ligado también a la posibilidad de utilizar más de un cluster de computadoras para

cómputo paralelo y en este contexto la sincronización de los relojes va más allá del análisis de rendimiento con el objetivo de optimizarlo

REFERENCIAS

[1] Agilent Application Note AN 1289, "The Science of Timekeeping"

[2] G. Coulouris, J. Dollimore, T. Kindberg, “Sistemas Distribuidos. Conceptos y Diseño”, 3ª edición. Pearson Educación, 2001. ISBN: 8478290494.

[3] F. Cristian. "Probabilistic Clock Synchronization". Distributed Computing, 3: 146–158, 1989.

[4] K. J. Elson, L. Girod, D. Estrin, "FineGrained Network Time Synchronization using Reference Broadcasts", Proceedings of fifth symposium on Operating System Design and Implementation. December 2002.

[5] C. Fetzer, F. Christian, "Integrating External and Internal Clock Synchronization", June 1996.

[6] D. A. Grove. "Performance Modelling of messagepassing parallel programs", May 2003.

[7] R. Gusella, S. Zatti "An Election Algorithm for a Distributed Clock Synchronization Program" EECS Department University of California, Berkeley Technical Report No. UCB/CSD-86-275 1986

[8] K. H. Kim, C. Im, P. Athreya, "Realization of a Distributed OS Component for Internal Clock Synchronization in a LAN Environment".Proc. ISORC 2002, IEEE 5th Int'l Symp on Objetoriented Realtime distributed Computing, Washington, D.C., April 2002, pp. 263270.

[9] D. L. Mills, "A Brief History of NTP Time: Confessions of an Internet Timekeeper". ACM Computer Communications Review 33, 2 (April 2003), pp 922.

[10] D. L. Mills, "Improved algorithms for synchronizing computer network clocks", IEEE/ACM Transactions on Networks June 1995.

[11] D. L. Mills, "Measured performance of the Network Time Protocol in the Internet System". ACM Computer Communication Review 20, Jan. 1990. pp. 6575.

[12] D.L. Mills, “Network Time Protocol (Version 3) specification, implementation and analysis”. Network Working Group Report RFC-1305, University of Delaware, March 1992, 113 pp.

[13] D.L. Mills, "Internet time Synchronization: the Network Time Protocol", IEEE trans. Communications COM39, October 1991, pp. 14821493.

[14] D. L. Mills, "Modelling and analysis of computer network clocks", Electrical Engineering Department Report 9252, University of Delaware, May 1992.

[15] D. L. Mills, “Precision synchronization of computer network clocks”, ACM Computer Communication Review 24, 2 (April 1994). 28-43.

[16] D. L. Mills, P. H. Kamp, "The Nanokernel", Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting (Reston VA, November 2000).

[17] D. L. Mills, "Unix kernel modifications for precision time synchronization". Electrical Engineering Department Report 94101, University of Delaware, October 1994.

[18] S. Mishra, C. Fetzer, F. Cristian, “The Timewheel Asynchronous Atomic Broadcast Protocol”, PDPTA 1997: 1239-1258

[19] F. L. Romero, W. Aróztegui, F.G. Tinetti, "Sincronización de Relojes en Ambientes Distribuidos" XII Congreso Argentino de Ciencias de la Computación (XII CACIC) Octubre 2006

[20] F. L. Romero, F. G. Tinetti, "Sincronización de Relojes en Ambientes Distribuidos" IX Workshop de Investigadores en Ciencias de la Computación,Trelew, Mayo 2007.

[21] D.B. Sullivan, D.W. Allan, D.A. Howe, and F.L. Walls, "Characterization of Clocks and Oscillators", NIST Tech Note 1337, 1990. (BIN: 868)

[22] P. Work, K. Nguyen, "Measure Code Sections Using The Enhanced Timer", http://www.intel.com/cd/ids/developer/asmo-na/eng/209859.htm, October 2005.

Provendo maior escalabilidade em HNOW com inclusão dinâmica de workers

Francisco Borges Santana
Centro de Pós-graduação e Pesquisa Visconde de Cairu - CEPPEV
Fundação Visconde de Cairu - Salvador, Bahia, Brasil
borgeskaiser@gmail.com

and

Josemar Rodrigues de Souza
Núcleo de Arquitetura de Computadores e Sistemas Operacionais
Universidade do Estado da Bahia - UNEB. Salvador, Bahia, Brasil
josemar@uneb.br

Abstract

Heterogeneous Networks of Workstations (HNOWs) constitute an efficient and cheap architecture which can provide greater computational power for scientific calculation. The HNOWs provides attractive scalability in terms of computational power and memory. One form to scale a cluster is to dynamically add more workstations, through libraries such as the MPI for example. However, to take off advantage of this feature, it's required the existence of a policy of load balancing that contemplates the dynamic addition of resources. It is necessary that the dynamic increase of workers either follow of a strategy of load balancing. The research considers, implements and evaluates a model of load balancing that deals with the dynamic inclusion of workstations in HNOW. The implementation is done using the MPICH2 in C/C++ with Master/Worker paradigm. For performance evaluation, we use the metric of speedup of the dynamic inclusions. We've made using as benchmark an algorithm of multiplication of matrix. The considered model provides a bigger scalability in HNOW with the dynamic inclusion of workers.

Keywords: Load balancing, Cluster Computing, Grid Computing, Master/Worker, Parallel algorithm.

Resumo

As Redes Heterogêneas de Estações de Trabalho (Heterogeneous Networks of Workstations - HNOW) constituem uma arquitetura eficiente e barata para prover maior poder computacional para cálculos científicos. As HNOWs proporcionam uma atrativa escalabilidade em termos de poder computacional e memória. Uma forma de escalar o cluster é adicionar mais estações de trabalho dinamicamente, através de bibliotecas como o MPI por exemplo. Contudo, para tirar proveito dessa facilidade, é necessário que exista uma política de balanceamento de carga que contemple a adição dinâmica de recursos. É necessário que o aumento dinâmico de workers seja acompanhado de uma estratégia de balanceamento de carga. A pesquisa propõe, implementa e avalia um modelo de balanceamento de carga que trata a inclusão dinâmica de estações de trabalho em HNOW. A implementação se faz utilizando o MPICH2 em C/C++ no paradigma Master/Worker. Para avaliação dos resultados são utilizadas a métrica speedup das inclusões dinâmicas. São efetuadas diversas medições utilizando como benchmark um algoritmo de multiplicação de matriz. O modelo proposto prove uma maior escalabilidade em HNOW com a inclusão dinâmica de workers.

Palavras chaves: Balanceamento de carga, Computação em Cluster, Computação em Grades, Master/Worker, Algoritmo paralelo.

1 INTRODUÇÃO

A possibilidade de dinamicamente incluir processos é um recurso poderoso e interessante para o desenvolvimento de aplicações paralelas. Essa característica, apesar de já existir no PVM, somente foi disponibilizada no MPI-2. Os fatores motivadores, para a inclusão dessa característica na especificação, foram o sucesso de Grid Computing e a necessidade de adaptar o comportamento dos programas paralelos durante a execução por conta da mudança de hardware [2]. A criação dinâmica processo é exemplifica por [8] com o balanceamento de carga, onde o algoritmo pode escolher a localização de onde os processos serão criados baseados em valores dinâmicos, onde a carga pode transparentemente migrar o processo de uma máquina para outra quando um recurso torna-se sobrecarregado ou indisponível. A exemplo da utilização da criação dinâmica de processo encontramos [2] que apresentam um modulo de scheduler, implementado em MPI-2, que determina em tempo de execução em qual processador um novo processo deve ser executado, provendo dessa forma um balanceamento de carga, outro exemplo que aplica essa mesma técnica é [6] que desenvolve um framework de balanceamento de carga. A criação dinâmica de processos também pode ser utilizada para a implementação de tolerância a falha [4]. O algoritmo paralelo pode perceber a indisponibilidade de determinado node e criá-lo em outro. Outra possibilidade da inclusão dinâmica de processos pode ser aplicada em HNOW - Heterogeneous Networks of Workstations , novos workers podem ser adicionados sob demanda para prover mais poder computacional a uma aplicação que iniciou o processamento com poucos workers. No entanto aumentar a quantidade de workers para ajudar no processamento de determinada tarefa não é o bastante para garantir que o algoritmo paralelo execute de maneira mais eficiente. É necessário que o aumento de workers seja acompanhado de uma estratégia de balanceamento de carga, pois a distribuição de trabalho é um fator crítico para o desenvolvimento de eficientes algoritmos paralelos [9]. O modelo é proposto para aplicações que implementam o paradigma Master/Worker com interações sincronizadas, o modelo ajusta a carga do cluster conforme a inclusão de novos workers. O modelo proposto foi desenvolvido para as HNOW, que utiliza o paradigma Master/Worker [1]. O modelo viabiliza o balanceamento de carga em HNOW, em que ocorrem a adição de workers dinamicamente, entende-se por inclusão dinâmica, a inclusão de novos workers no cluster sem a interrupção do processamento.

2 MODELO CONCEITUAL

As HNOW são caracterizadas pelo fato dos workers possuírem recursos computacionais heterogêneos. [11] apresentam diversos fatores de heterogeneidade como arquitetura da máquina, latência de rede, largura de banda, memória, entre outros. Além disso, as HNOW geralmente não são dedicadas para a execução do algoritmo paralelo. Por isso, a capacidade de processamento dos workers é diferente e sofre variações ao longo do cômputo, influenciando a performance do algoritmo. Assim, é necessário que a quantidade de tarefas executadas por determinado worker seja proporcional à sua capacidade de processamento para evitar que todo o cluster fique aguardando pelo término do processamento do worker mais lento. A distribuição das tarefas está relacionada também com a quantidade de workers no cluster. Se a quantidade de workers aumentar a quantidade de tarefas enviadas para os outros workers deve ser reavaliada. A figura

1 apresenta o modelo proposto.

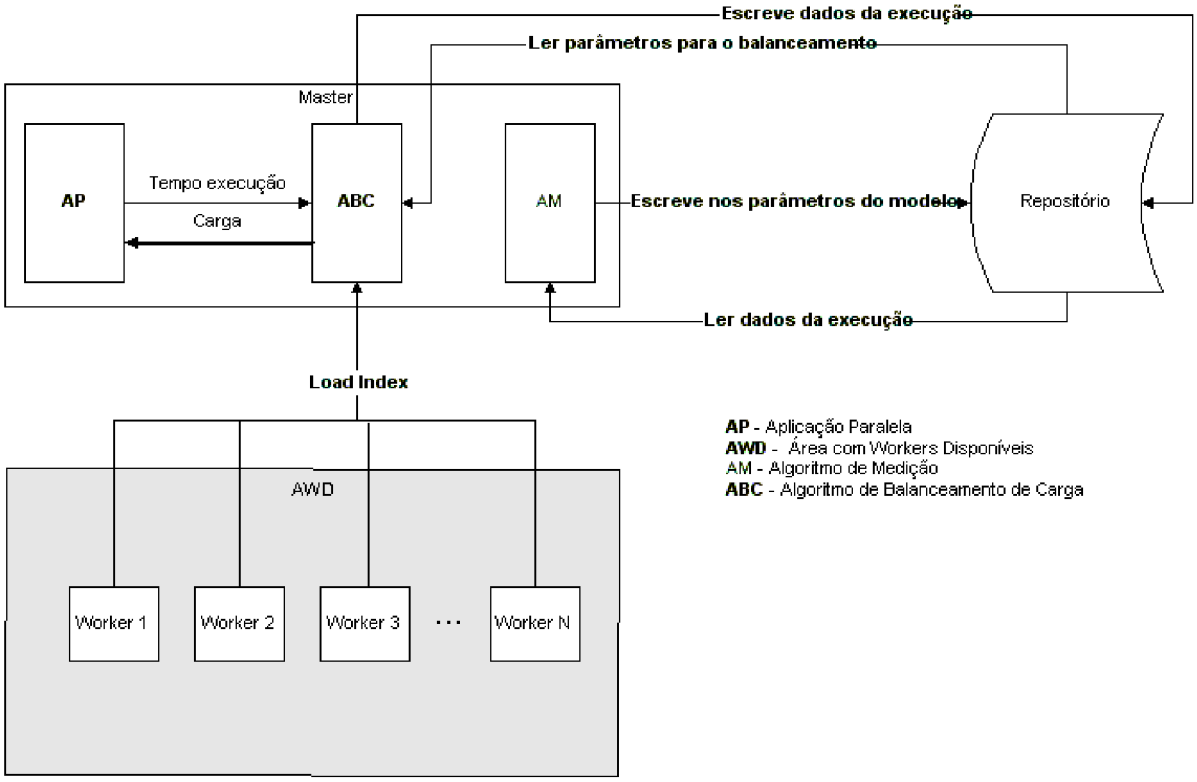


Figura 1: Modelo proposto

O modelo é composto: AWD - Área de Workers Disponíveis, AP - Algoritmo Paralelo, ABC -Algoritmo de Balanceamento de Carga e o AM -Algoritmo de Medição. O repositório é o local onde o master recupera e armazena informações sobre o processamento do algoritmo paralelo e do modelo. O repositório possui os parâmetros de configuração do algoritmo de balanceamento. Um desses parâmetros é o THRESH- OLD MAX que indica, em percentual, o limite superior que o tempo de computo de um worker deve ter. Outro é o THRESHOLD MIN - indica, em percentual, o limite inferior que o tempo de computo de um worker deve ter. O tamanho do bloco inicial que será utilizado para cálculo é indicado pelo parâmetro VALTAM BLOCO. O parâmetro WORKERS ADD que indica quanto workers serão adicionados dinamicamente. E por fim o PERC CONS ADD indica quando, em percentual de dados processados, os workers serão adicionados dinamicamente.
Quando o master executa o algoritmo de balanceamento de carga, as informações contidas no repositório são lidas e passadas como parâmetro para o algoritmo, que efetua os cálculos necessários para a distribuição de carga. Os resultados obtidos após execução são armazenados no repositório. É através do repositório que o algoritmo de balanceamento de carga consegue perceber a existência de recursos computacionais adicionados no cluster.
A Área de Workers Disponíveis é um recurso computacional que possui todos os workers que podem ajudar a executar determinado processamento. Essa área pode compartilhar diversos workers, que podem fazer em dado instante parte de um cluster, que processa um algoritmo A, por exemplo, e em outro momento fazer parte de outro cluster, que processa o algoritmo B. Esse recurso permite a adição

dinâmica de worker e isola a aplicação paralela e o algoritmo de balanceamento de carga, de forma que a manipulação dessa área não os afeta. O algoritmo paralelo deve possuir dados que possam ser divididos em diversas tarefas e cada tarefa deve possuir as mesmas características que a outra.

O algoritmo de balanceamento de carga é iniciado quando o master sincroniza as tarefas enviadas para os workers. Nesse momento, o algoritmo de balanceamento efetua os cálculos necessários e envia a carga apropriada de cada worker. Conforme [10], todo trabalho executado no paradigma master/worker é iniciado quase simultaneamente. Logo, a alocação de tarefas deve ser proporcional à capacidade computacional [3], para permitir que a execução de todos os workers em cada interação termine aproximadamente no mesmo tempo. Com as informações obtidas do repositório e com as informações enviadas dos workers sobre a performance(load index), o algoritmo de balanceamento faz a predicação da carga que deve ser distribuída. Em ambientes homogêneos e dedica- dos, os recursos computacionais, tais como, memória disponível, capacidade de processador, largura de rede e latência de rede são conhecidos, diferentemente de ambientes heterogêneos e não dedicados, onde esses valores variam constantemente. Uma forma de se obter esses valores é através da predicação. Conforme [12], predicar a futura performance baseada em informações do passado é uma abordagem comum, feita para balanceamento de carga em HNOW.

O reajuste de carga é efetuado, quando o tempo de computo (T_{comp}) dos workers está fora do intervalo definido pelo threshold superior e inferior em determinada interação. O valor do threshold é calculado sobre o valor do pior tempo de execução obtido em cada interação. Assim como salientam [10], levar-se-´a em consideração o pior caso em que o tempo de comunicação e o de computação não são sobrepostos. Dessa forma, pode-se identificar qual parâmetro de heterogeneidade está influenciando no tempo de execução da interação: tempo de comunicação ou tempo de computação. A carga somente é ajustada para aqueles workers cujo tempo de computação $T_{comp,i}$ não esteja no intervalo dos thresholds inferior e superior, previamente definido. Os thresholds inferior e superior são um percentual sobre o tempo previsto para a execução de uma interação. Se o $T_{comp,i}$ estiver abaixo do threshold inferior, indica que o worker está subutilizado. Acima do threshold superior, indica uma sobrecarga. Entre esses intervalos, indica que a carga está em uma situação ideal. Portanto, nota-se que o tempo de computação da tarefa $T_{comp,i}$ é dependente do processador e/ou da memória disponível no worker para a execução da tarefa. Logo, infere-se, que se o tempo de computação for maior que o tempo previsto para a interação, há uma sobrecarga de trabalho ou o worker, não possui recursos Disponíveis, no momento, para atender a requisição de forma eficiente. O tempo de execução de uma tarefa, porém, depende da aplicação paralela e da heterogeneidade do ambiente. Por isso, a necessidade de se caracterizar cada worker na primeira interação. A caracterização consiste no master enviar uma tarefa com as mesmas características para ser processada, a fim de obter a capacidade computacional do worker. O pior tempo de computação obtido será considerado como o tempo necessário para executar uma tarefa. Assim, a performance ficará limitada ao worker mais lento. A quantidade de tarefas enviadas para cada worker na primeira interação é definida pela Eq. 1.

$$QtarefaW = TAM\ BLOCO * FatorHet_i \quad (1)$$

Onde TAM_BLOCO é o tamanho do inicial enviado para todos os workers o qual é definido em arquivo de configuração do modelo e $FatorHet_i$ é o fator de heterogeneidade do worker que receberá a carga. Como trata-se da primeira interação esse valor é igual a 1 para todos os workers. No entanto esse fator de heterogeneidade pode variar em cada interação e pode variar em diferentes aplicações paralelas. O fator de heterogeneidade, representado na equação 2, é calculado dividindo a quantidade de blocos enviados na interação anterior sobre a quantidade de blocos inicial. Por isso esse fator é sensível aos recursos computacionais disponíveis, se um worker processar mais em uma interação significa que seu fator de heterogeneidade será maior na próxima interação, logo receberá mais trabalho, o contrário ocorre caso o fator de heterogeneidade seja menor. Além disso, esse fator retrata a diferença computacional existente entre os workers. A tendência é que esse valor seja uma constante durante a execução do algoritmo. Nem sempre a quantidade de tarefas que deve ser enviada será um número inteiro, nesses casos efetua-se um arredondamento para a parte inteira.

$$FatorHet_i = \frac{FatorHet,i\ Anterior}{TAM\_BLOCO} \quad (2)$$

Cada worker irá processar Ns tarefas conforme sua capacidade de processamento e disponibilidade de recursos - processador e memória. A quantidade de tarefas executadas por cada worker a partir da segunda interação é definida no momento da sincronização, seguindo os critérios abaixo:

1. Se o tempo de computação for menor que o threshold inferior, aumenta a quantidade de tarefas para o worker, indicando que está subutilizado.

$$QtarefaW = ((\frac{Thresholdinferior}{Tcomp,i}) * TAM\_BLOCO) * FatorHet,i \quad (3)$$

2. Se o tempo de computação for maior que o threshold superior, diminui a quantidade de tarefas para o worker, indicando que está sobrecarregado.

$$QtarefaW = ((\frac{Thresholdsuperior)}{Tcomp,i}) * TAM\_BLOCO) * FatorHet,i \quad (4)$$

3. Se o tempo de computação estiver entre o threshold inferior e o superior, a quantidade de tarefas para o worker permanece a mesma que a anterior, indicando que está com o balanceamento de carga ideal.

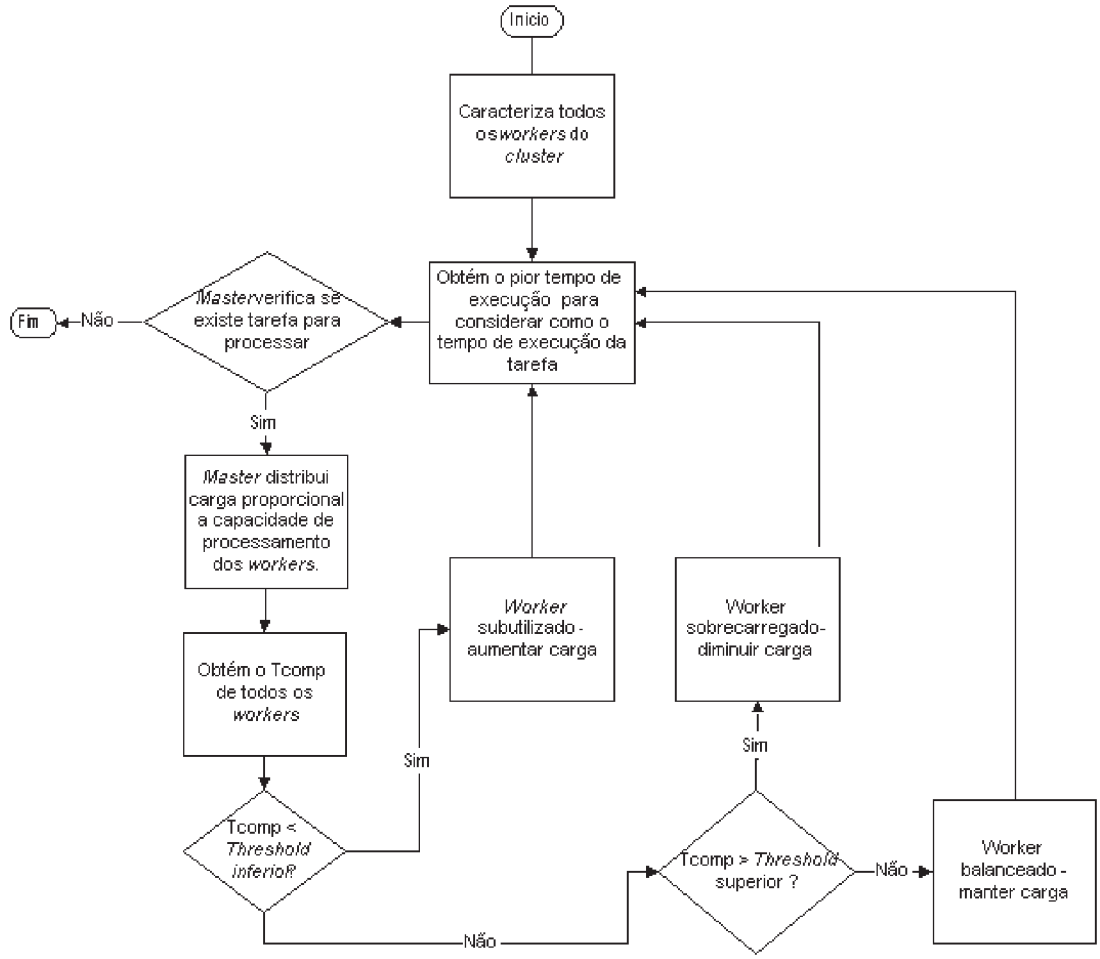
$$QtarefaW = TAM\_BLOCO * FatorHet,i \quad (5)$$

Onde $QtarefaW$ indica a quantidade de carga que será enviada para o worker; threshold inferior e threshold superior indicam os valores em ms dos threshold inferior e superior respectivamente; Tcomp,i representa o tempo de computo da interação anterior do worker i e $FatorHet,i$ indica o fator de heterogeneidade do worker i. Em cada interação, o algoritmo de balanceamento de carga verifica qual é a possível quantidade de tarefas que os workers têm capacidade de processar.
Cada worker, após o término do processamento de cada tarefa, informa ao algoritmo

de balanceamento o load index - tempo de computação e tempo de comunicação. [7] afirma que um load index comum utilizado em muitos algoritmos paralelos é a diferença entre o tempo que o worker leva para iniciar e finalizar uma tarefa particular.

O fluxograma do algoritmo de balanceamento de carga é apresentado na Fig. 2.

Figura 2: Fluxograma do algoritmo de balanceamento de carga



As estratégias de distribuição de bloco existentes, não atendem aos requisitos do modelo proposto, por conta disso é necessária a proposta de outra estratégia de distribuição. O modelo proposto necessita que a carga enviada seja dinâmica e varie conforme performance do worker envolvido no processamento. A carga enviada pode ser diferente entre os workers e diferentes para o mesmo worker em interação diferente. A estratégia proposta é uma variação na estratégia de distribuição conhecida como Block- cyclic distribution. A Block-cyclic dinamic distribution, estratégia proposta exibida na figura 3, consiste em variar o tamanho do bloco que é utilizado na Block-cyclic distribution, nessa distribuição o bloco tem tamanho fixo e é enviado ciclicamente para os processo. Na Block-cyclic dinamic distribution o tamanho do bloco varia conforme a capacidade de

processamento do worker que é obtido pelo load index, Além de ser enviado ciclicamente também.

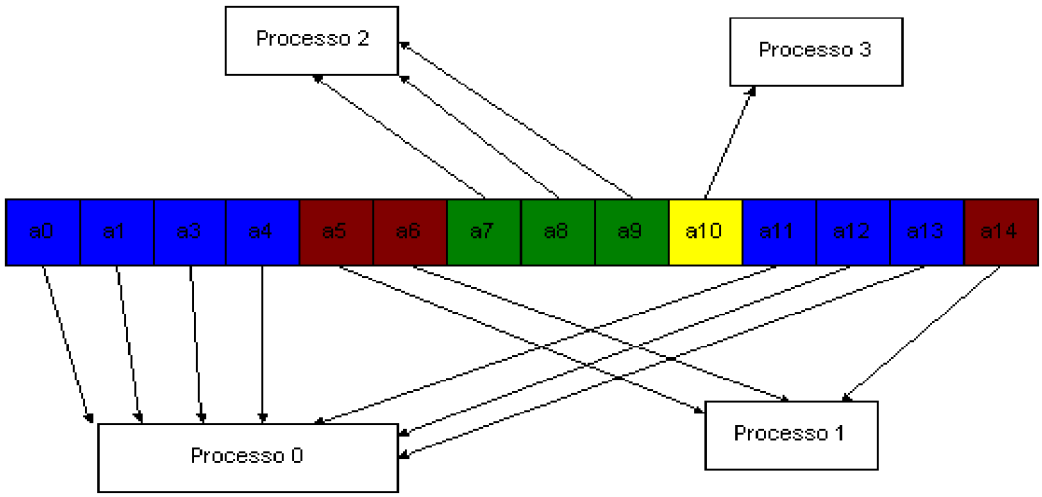


Figura 3: Block-Cyclic Dinamic Distribution de 15 blocos distribuídos entre 4 processos com tamanho de bloco variando conforme performance do processo.

Na Block-cyclic dinamic distribution, é possível que algum processo fique sem receber trabalho na última interação. A razão disso é porque o envio de dados para os processos ocorre de forma dinâmica e com a variação no tamanho da carga enviada. Porém acredita-se que os benefícios obtidos pela distribuição sobreponham esse fato.

O algoritmo de Medição tem a função de verificar se a estratégia de balanceamento está sendo eficiente e se os parâmetros informados no modelo estão apropriados para o ambiente, sugerindo resultados que permitam ajustar os parâmetros de maneira adequada. O algoritmo de Medição para efetuar essa verificação, obtém as informações sobre a execução do algoritmo de balanceamento que é registrado no repositório e da saída da aplicação paralela. Com essas informações pode-se inferir a eficiência dos parâmetros utilizados no modelo.

3 IMPLEMENTAÇÃO DO MODELO CONCEITUAL

A linguagem de programação utilizada foi o C/C++, a implementação do MPI utilizada foi o MPICH2 [8]. A principal razão para a utilização do MPICH2 é porque essa implementação separa os processos de comunicação e de gerenciamento, sendo o ambiente de execução padrão composto por um conjunto de daemons chamados de MPD - Managing Process Daemon que estabelece a comunicação entre as máquinas antes do processo da aplicação inicializar [5], essa característica permite facilmente a implementação do AWD. A aplicação paralela desenvolvida para ser utilizada no modelo foi a multiplicação de matriz quadrada. Considerando as matrizes quadradas A = (aij) e B = (bij) de ordem n, então C = (cij) = AB é uma matriz quadrada de ordem n

e cij é obtido efetuando a multiplicação da linha ith de A pela coluna jth de B conforme a equação 6 .

$$cij = ai0b0j + ai1b1j + \ldots + ai,n-1bn-1,j \quad (6)$$

Apesar do algoritmo de Medição ser proposto no modelo, o mesmo não foi implementado, pois o propósito do mesmo é analisar os resultados gerados pelo modelo para sugerir parâmetros otimizados para o modelo, como por exemplo o tamanho do bloco inicial, o threshold inferior e superior. A implementação desse algoritmo pode- ria ser utilizando-se algumas técnicas como inteligência artificial, estatística, mineração de dados ou qualquer outra que pudesse efetuar alguma aferição. A Medição foi efetuada utilizando-se da observação dos valores obtidos durante as medições.

4. CARACTERIZAÇÃO E NORMALIZAÇÃO DO CLUSTER

O Cluster utilizado para obter os resultados da pesquisa foi composto por 6 máquinas, as quais são descritas na tabela 1.

Tabela 1: Caracterização do Cluster

Nome	Configuração das máquinas
master	AMD Athlon 64 bits 2.210 Mhz 1 Gb RAM 5Gb HD
worker01	Digital Prioris MX 6200, 256 MB RAM, 2 Proc. Pentium Pro200Mhz HD SCSI
worker02	Digital Server 3000, 512 MB RAM, Pentium II 266Mhz SCSI
worker03	Digital Prioris MX 6200, 256 MB RAM, 2 Proc. Pentium Pro 200Mhz SCSI
worker04	AMD-K6 II 500 Mhz 256 Mb RAM HD IDE 4.3 Gb
worker05	Pentium II 133 MHz 128 MB RAM HD IDE 4 GB

Como pode ser visto na tabela 1 o cluster é heterogêneo, logo a capacidade de processamento entre as máquinas são diferentes. Por isso, para a análise da efficiency, não se pode considerar como a quantidade de processos envolvidos no cálculo a quantidade física de máquinas no cluster, é necessário normalizar a capacidade de processamento entre essas máquinas. Normalizar significa equiparar a capacidade de cada máquina em relação à mais potente. Para identificar a máquina mais potente executou-se o beackmark de multiplicação de matriz serial para uma matriz quadrada de 600 em todos os workers. Os tempos obtidos são listados na tabela 2.

Tabela 2: Tempos obtidos com *beackmark*

Worker	Tempo obtido(s)
Worker01	29,448000
Worker02	28,964000
Worker03	29,450000
Worker05	83,812000
Worker04	111,390000

No beackmark o melhor tempo obtido foi do worker02, esse tempo é usado como dividendo nos outros tempos para obter-se o valor normalizado. A exemplo, o valor nor- malizado do worker04 é 0,260023 que corresponde a 28,964000 dividido por 111,390000.
A tabela 3 apresenta os valores normalizados do cluster.

Tabela 3: Normalização do Cluster

Worker	Tempos beackmark(s)	Melhor tempo obtido(s)	Valor normalizado
Worker01	29,448000	28,964000	0,983564
Worker02	28,964000	28,964000	1
Worker03	29,450000	28,964000	0,983497
Worker05	83,812000	28,964000	0,345583
Worker04	111,390000	28,964000	0,260023

5. SPEEDUP DO ALGORITMO DE BALANCEAMENTO DE CARGA

Para a validação do modelo foi verificado o speedup do algoritmo sem a inclusão dinâmica de workers. A figura 4 mostra que o algoritmo de balanceamento de carga sem a inclusão dinâmica apresenta-se escalável pois, com o aumento de workers no cluster houve um aumento nos speedups. Acredita-se que pode-se obter speedups melhores ajustando o tamanho do bloco e o parâmetro do threshold através implementação do algoritmo de Medição.

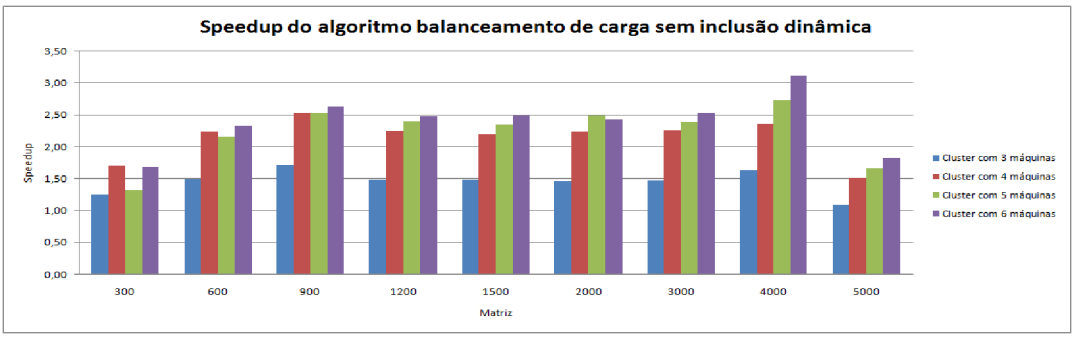


Figura 4. Speedup do algoritmo de balanceamento de carga sem inclusão dinâmica.

O algoritmo apresentou speedups crescentes com o aumento da quantidade de workers no cluster, com exceção das matrizes de 300, 600 e 2000. Na matriz de 300 o speedup para o cluster formado por cinco e seis máquinas foram inferiores a do cluster formado por quatro e na matriz de 600 o speedup para o cluster formado por cinco máquina foi inferior a do cluster formado por quatro. As razões para esse comportamento são:

1. Etapa da caracterização - na primeira interação o algoritmo precisa verificar a capacidade computacional de cada worker, para isso é enviado uma carga de trabalho igual para todos os workers, assim workers com poder

computacional diferentes recebem a mesma quantidade de tarefas. Essa operação tem impacto no tempo total de execução quando a matriz não é relativamente esparsa, como por exemplo, as de 300 e 600.

2. Heterogeneidade dos workers - a normalização indica o quanto um worker equivale ao worker mais potente cluster. Dependendo do índice de normalização do workers que fazem parte da caracterização o impacto será maior ou menor no tempo total de execução. Quanto maior for a heterogeneidade maior será o tempo de processamento do worker com menos poder computacional no momento da caracterização. Observa-se na própria matriz de 300 quando o cluster é formado por três e quatro máquinas que o speedup foi crescente, pois o índice de normalização para o worker03 que compõe o cluster formado por quatro máquinas é de 0,983497 sendo 1 e 0,983564 o índice do workers02 e worker01 respectiva- mente, dessa forma o poder computacional desses workers são bem próximos, diferentemente do worker05, cujo índice é de 0,345583.
3. Overhead - quanto maior a quantidade de worker presentes no cluster maior será o overhead o ocorrido nas interações do algoritmo para o envio, recepção e outros cálculos efetuados pelo algoritmo. Quanto mais esparsa for a matriz menos significante será a influência desses cálculos no tempo total de execução do algoritmo.

A outra verificação efetuada para a validação do modelo foi a execução do algoritmo de balanceamento de carga com a inclusão dinâmica. Na figura 5 pode-se verificar, com exceção da matriz de 300, que todos os speedups obtidos foram superiores a curva do cluster formado por 3 máquinas. Nessa configuração a inclusão dinâmica de um worker a partir da matriz de 300 já fornece tempos execuções melhores.

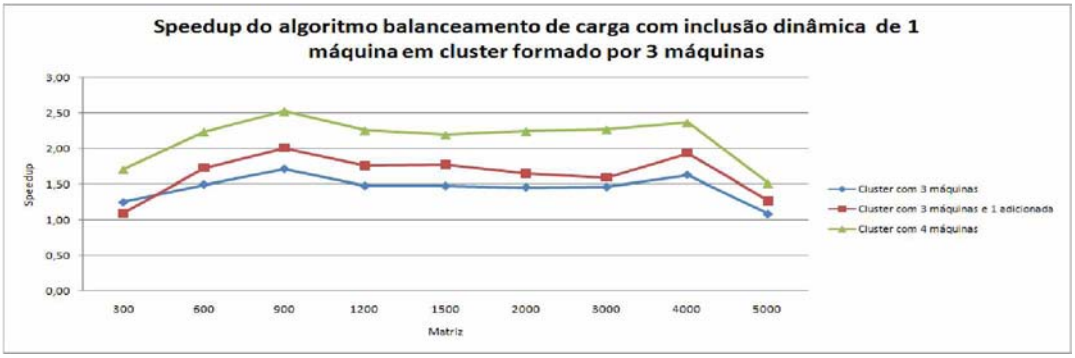


Figura 5: Speedup do algoritmo de balanceamento de carga com inclusão dinâmica de uma máquina em Cluster formado por três.

Na figura 6 pode-se verificar que os speedups obtidos com a inclusão dinâmica tem speedups crescentes a partir da matriz de 300, no entanto, esses speedups somente são superiores a curva do cluster formado por 3 máquinas depois da matriz de 2000. Nessa Medição houve speedup superior as curvas do cluster com quatro e cinco máquinas.

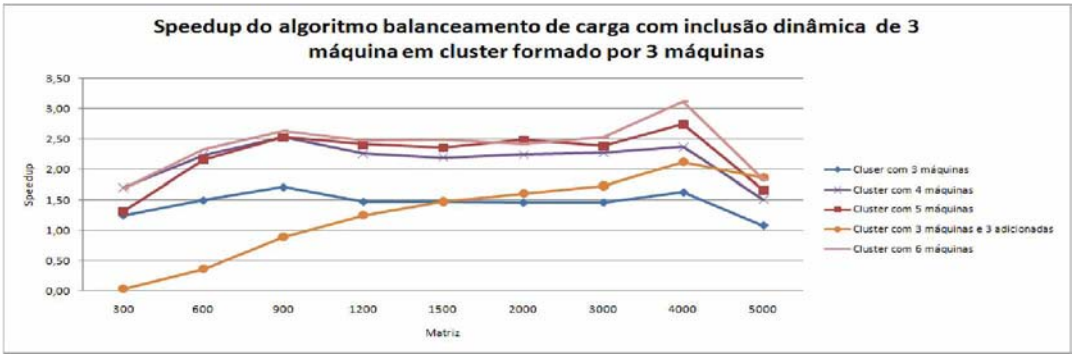


Figura 6: Speedup do algoritmo de balanceamento de carga com inclusão dinâmica de 3 máquinas em Cluster formado por três.

A figura 7 apresenta um gráfico com todos os speedups obtidos. Pode-se observar nas curvas que os piores tempos, foram obtidos quando três workers foram adicionado em um cluster formado por três máquinas. Aferiu-se quanto tempo era gasto para criar processos remotamente e verificou-se que esse tempo varia de máquina para máquina. Sendo assim, o tempo para criar um processo remoto no worker03 é menor do que criar no worker05 e criar no worker05 por sua vez, leva menos tempo do que o worker04. Por isso a quantidade de workers adicionados e qual worker foi adicionado, influencia diretamente no tempo de execução.

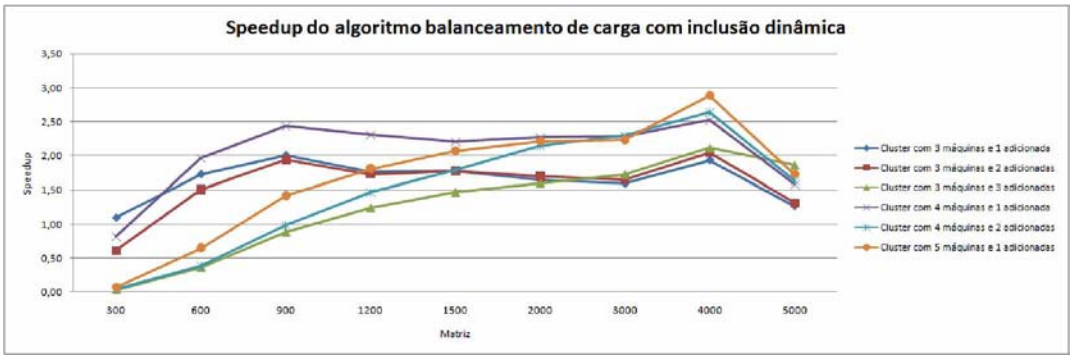


Figura 7: Speedup do algoritmo de balanceamento de carga com inclusão dinâmica.

Não se pode afirma que quanto maior a quantidade de workers a serem adiciona- dos menor será o speedup obtido, pois depende do custo da criação do processo remoto, o qual depende do worker a ser adicionado. Nas medições o decréscimo dos speedups foram observados com o aumento da quantidade de workers adicionados, porém isso ocorreu porque a ordem de inserção sempre foi worker03, worker05 e worker04, cujo custo de adição remota de processo aumenta. Outra evidência da influencia do tempo na criação de um processo remoto pode ser constatada nas curvas ”Cluster com 5 máquinas e 1 adicionada”e ”Cluster com 4 máquinas e 1 adicionada”. Os speedups apresentados por essa curva são melhores do que a outra, apesar de possuir menos máquinas, pois a máquina adicionada na primeira curva é o worker04 já na segunda é o worker05.

6 CONCLUSÃO

A pesquisa propôs, implementou e avaliou um modelo para balanceamento de carga em HNOW que contempla a inclusão dinâmica no cluster em aplicações master/worker. Em HNOW o principal fator desafiador é a heterogeneidade do ambiente. O modelo apresentado trata como fatores de heterogeneidade a capacidade de processamento de cada worker - enviando para cada worker um trabalho correspondente a sua capacidade; e a inclusão dinâmica de workers que é ajustar toda a carga do cluster conforme capacidade de processamento do worker adicionado. O algoritmo de balanceamento de carga foi avaliado com e sem a inclusão dinâmica de workers. Nas duas avaliações o algoritmo distribuiu a carga conforme capacidade de processamento dos workers próximo aos valores normalizados de cada worker, atingindo assim o seu propósito. Os speedups obtidos do algoritmo de balanceamento de carga sem a adição dinâmica quase em todas as medições foram maiores quando o cluster era formado por uma quantidade maior de workers. Isso indica que o algoritmo apresentou-se escalável sem a inclusão de novos workers. Os speedup obtidos do algoritmo de balanceamento de carga com a adição dinâmica também apresentou-se escalável, porem não em todas medições, indicando que a escalabilidade depende da capacidade, normalizada, do worker adicionado. Percebe-se que a razão para isso são os tempos gastos para a criação e gerenciamento remoto de processos. O trabalho constatou que é possível obter escalabilidade do cluster HNOW utilizando-se de criação dinâmica de processo para adicionar novos workers. Pretende-se, como trabalho futuro, ajustar e implementar o modelo em multi-cluster.

REFERÊNCIAS

[1] Rajkumar Buyya. High Performance Cluster Computing: Architectures and Systems, volume 1.Prentice Hall PTR, New Jersey, USA, 1999.

[2] Márcia C. Cera, Guilherme P. Pezzi, Elton N. Mathias, Nicolas Maillard, and Philippe Olivier Alexandre Navaux. Improving the dynamic creation of processes in mpi-2. In Bernd Mohr, Jesper Larsson Traff, Joachim Worringen, and Jack Dongarra, editors, PVM/MPI, volume 4192 of Lecture Notes in Computer Science, pages 247–255. Springer, 2006.

[3] Mark A. Franklin and Vasudha Govindan. A general matrix iterative model for dynamic load balancing. Parallel Computing, 22(7):969–989, 1996.

[4] G. A. Geist, J. A. Kohla, and P. M. Papadopoulos. Pvm and mpi: A comparison of features. Calculateurs Paralleles, 8(2):137–150, 1996.

[5] William Gropp, Ewing Lusk, David Ashton, Darius Buntinas, Ralph Butler, Anthony Chan, Rob Ross, Rajeev Thakur, and Brian Toonen. Mpich2 user's guide. Technical report, Mathematical, Information, and Computational Sciences Division, Novem- ber 2005.

[6] Kaoutar El Maghraoui, Travis Desell, Boleslaw K. Szymanski, James D. Teresco, and Carlos A. Varela. Towards a middleware framework for dynamically reconfigurable scientific computing. In L. Grandinetti, editor, Grid Computing and New Frontiers of High Performance Processing, volume 14 of Advances in Parallel Computing, pages 275–301. Elsevier, 2005.

[7] Shahzad Malik. Dynamic load balancing in a network of workstations. Technical Report 219762, Carleton University, November 2000.

[8] Message Passing Interface Forum MPIF. Mpi-2: Extensions to the message-passing in- terface. Technical Report, University of Tennessee, Knoxville, 1996.

[9] Ashraf Osman. Designing a scalable dynamic load-balance algorithm for pipelined sin- gle program multiple data applications on a non-dedicated heterogeneous network of workstations. Phd thesis, West Virginia University, Morgantown, West Virginia, USA, 2003.

[10] Gary Shao, Rich Wolski, and Fran Berman. Performance effects of scheduling strategies for master/slave distributed applications. Technical Report CS98-598,

University of California, San Diego, 1998.

[11] Mohammed J Zaki, Wei Li, and Michal Cierniak. Performance impact of processor and memory heterogeneity in a network of machines. Technical Report 574, Department of Computer Science, University of Rochester, 1995.

[12] Mohammed Javeed Zaki, Wei Li, and Srinivasan Parthasarathy. Customized dynamic load balancing for a network of workstations. Journal of Parallel and Distributed Computing, 43(2):156–162, 1997.

Reconocimiento de patrones de tráfico de red en un ambiente Condor

Paula Martínez, Carlos Catania y Carlos García Garino
LAPIC, Instituto Tecnológico Universitario, Universidad Nacional de Cuyo
Mendoza, 5500, Argentina
pmart@uncu.edu.ar, {ccatania,cgarcia@itu.uncu.edu.ar}

y

Javier Díaz
LINTI, Facultad de Informática, Universidad Nacional de la Plata
Calle 115 entre 49 y 50, La Plata, Argentina
jdiaz@unlp.edu.ar

Resumen

La clave de la computación de alta disponibilidad es hacer un uso eficiente de los recursos disponibles. En este trabajo se describen las características y funcionalidades del software Condor que permite la administración y explotación eficiente de los recursos en dichos entornos.
Se discute la ejecución en un ambiente Condor de una aplicación diseñada para obtener patrones de tráfico de red. Además, presenta la infraestructura del sistema sobre la que se ejecuta dicha aplicación concurrente. La ejecución de la aplicación sobre el ambiente Condor no sólo ha permitido mejorar el tiempo total de ejecución sino también obtener mejores resultados respecto del caso serial. Pese a la pequeña infraestructura utilizada, las mejoras obtenidas muestran la capacidad de Condor como un sistema por lotes de computación distribuida de alta disponibilidad.

Palabras clave: Condor – HTC – IDS – algoritmos genéticos – tráfico de red

1. INTRODUCCIÓN

En la actualidad la potencia de cálculo es un factor a tener en cuenta en la realización de todo proyecto de investigación que requiere recursos computacionales. En muchos casos se necesita un entorno computacional que brinde una gran capacidad de cómputo durante largos períodos de tiempo.
Gracias al incremento de la potencia de los computadores y a las tecnologías de computación distribuida es posible atacar problemas difíciles de resolver un tiempo atrás (debido al tiempo necesario o al costo de la infraestructura de computación a emplear). Se pueden distinguir dos paradigmas de empleo de un cluster:

- HPC (High Performance Computing): En este paradigma se prima la ejecución lo más rápido posible de las tareas. Conceptos tales como paralelización y multiproceso entran dentro de este ámbito, y su aplicación directa es hacer que cálculos que pueden durar semanas en un solo equipo se repartan entre varios, dividiendo el trabajo a realizar.

- HTC (High Throughput Computing): En este paradigma se prima la ejecución de la mayor cantidad posible de tareas. Conceptos como gestión de colas y de recursos son parte de este ámbito, y su aplicación directa pasa por la realización de la mayor cantidad de trabajos a lo largo del tiempo.

El paradigma de HTC puede utilizarse con facilidad cuando se tiene una multitud de cálculos de tamaño mediano / pequeño, y se quiere agilizar la gestión de todos ellos [1].
Condor [2] es un entorno distribuido diseñado para Computación de Alta Disponibilidad (High Throughput Computing HTC) y empleo de recursos ociosos. Este permite de manera transparente y simultánea explotar las capacidades de estaciones de trabajo que pueden estar distribuidas en el mundo y pertenecer a distintos individuos, grupos, departamentos e instituciones. La idea principal de Condor es hacer posible que cálculos con alto costo de gestión puedan ser realizados de forma ágil y eficiente.
En este trabajo se discuten los cambios necesarios para la adaptación al entorno Condor de una aplicación de reconocimiento de patrones en el tráfico de red basada en algoritmos genéticos [3]. Debido a la característica no determinística de las aplicaciones basadas en algoritmos genéticos, surge la necesidad de repetir su ejecución numerosas veces con el fin de realizar las validaciones estadísticas. La característica fuertemente desacoplada de este tipo de validaciones estadísticas y el alto consumo de recursos computacionales que las mismas demandan, sugieren su ejecución en ambientes distribuidos como Condor.
El trabajo se organiza de la siguiente manera: en la segunda sección de este se describe la infraestructura para ejecutar trabajos con Condor. Posteriormente, en la sección tres se detallan las características más interesantes de Condor y su funcionamiento general.
La gestión y administración de Condor, que incluye tareas como el acceso de los usuarios al conjunto de máquinas administradas por Condor, los mecanismos de autenticación y autorización de usuarios, las políticas de ejecución de trabajos, el mecanismo de checkpointing, etc., se discuten en la sección cuatro.
En la sección cinco se describen los archivos de emisión de los trabajos, los entornos de ejecución (universos) que soporta Condor y se ejemplifica la ejecución de un trabajo en Condor sobre la infraestructura disponible. La aplicación para reconocimiento de patrones en el tráfico de red se discute en la sección seis y finalmente se brindan las conclusiones en la sección siete.

2. INFRAESTRUCTURA

En esta sección se describe la infraestructura disponible para ejecutar trabajos de Condor El mismo incluye el cluster Reloaded y la computadora llamada Glapic (ver Figura 1) que se comporta como front end.
Glapic posee un P4 HT de 3.0 GHz, 1 MB de Cache L2, 1 GB de RAM, disco rígido de 160 GB, y un adaptador de red Fast Ethernet.
El cluster Reloaded tiene una estación de trabajo maestra que actúa como front end y 12 estaciones de trabajo esclavas. Estas estaciones conforman lo que se conoce como pool en la nomenclatura de Condor.
Las estaciones poseen un procesador P4 HT 3.0 GHz, 1 MB de Cache L2, 1 GB de RAM, disco rígido SATA de 80 GB, y un adaptador de red Gigabit Ethernet. La computadora maestra y las estaciones de trabajo del cluster están interconectadas mediante una red Gigabit Ethernet.

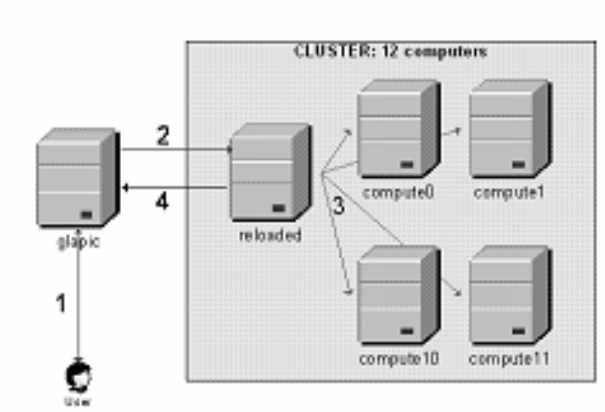


Figura 1. Cluster Reloaded

Glapic cuenta con Fedora Core 3 Kernel 2.6.9 como sistema operativo y con Condor 6.8.4 como middleware. El cluster Reloaded cuenta con la distribución para cluster Rocks 4.1.

3. CARACTERÍSTICAS Y FUNCIONAMIENTO DE CONDOR

Algunas de las características más interesantes de Condor son las siguientes:

- Sistema de colas integrado. El trabajo se envía a Condor. El entorno escoge el nodo más adecuado y lo ejecuta.
- Capacidad de ejecutar una simulación numerosas veces sin tener que relanzar a mano el cálculo.
- Notificación de finalización del cálculo vía e-mail.
- Posibilidad de realizar selección (es posible decir "solo lances el cálculo en máquinas con más de 512 MB de RAM).
- Posibilidad de preferencias ("lanza el cálculo en la máquina que tenga más HD").
- Sistema de prioridades sencillo (en el usuario y en el administrador).
- Configuración muy sencilla (trabaja una vez, y reusa).
- Funciona en nodos no dedicados, sin molestar al usuario.
- Disponible en Windows y Linux (es posible dentro de una configuración que el trabajo se ejecute en multiplataforma).
- Posibilidad de checkpointing (se guarda cada cierto tiempo el estado del cálculo).
- Conexión a Globus directa: Es posible emplear la potencia de cálculo de otros cluster mediante Condor-G.
- Posibilidad de ejecución condicional.

Condor realiza dos funciones principales: procesado de colas y gestión de recursos. El procesado de colas consiste en recoger las peticiones de trabajos y ponerlas de forma priorizada en una cola para su ejecución, mientras que la gestión de recursos se encarga de saber qué equipos están activos y con qué características cuentan, así como de repartir los trabajos de forma óptima.
Para recoger los trabajos Condor emplea un fichero de configuración que tiene todos los parámetros necesarios del mismo (programa a ejecutar, ficheros de configuración, ficheros de salida, variables de entorno, etc.). Dicho fichero de configuración debe de ser generado por el usuario mediante una sintaxis sencilla y potente (existen ficheros base explicados, así como ejemplos para las configuraciones más frecuentes).

Condor recoge todos los trabajos a ejecutar y examina los recursos existentes en el sistema. Si hay equipos suficientes para todos los trabajos los ordena y envía directamente. Si no hay equipos suficientes, los ordena según prioridades y los va ejecutando a medida que quedan equipos libres.
La gestión de recursos se realiza mediante unas etiquetas asignables tanto a los trabajos como a los recursos, que se denominan ClassAds y especifican las características de unos y otros. Una vez realizada esta caracterización, la gestión de recursos únicamente consiste en asociar los ClassAd de trabajos con los de recursos.
Una vez que Condor decide ejecutar un trabajo, transfiere tanto el programa como los ficheros necesarios para su ejecución al recurso adecuado y comienza su ejecución. Los resultados (así como un fichero de log en el que se refleja el progreso de la tarea) se transfieren de nuevo al equipo del usuario.
Condor permite el checkpointing, que consiste en guardar de forma periódica el estado completo de una tarea para poder recuperarla a posteriori.

4. ADMINISTRACIÓN DE RECURSOS

En esta sección se detalla la forma en que Condor administra la ejecución de trabajos, como resuelve el encolamiento, planificación y checkpointing de los trabajos, la asociación de solicitudes de recursos con las ofertas de recursos y aspectos de seguridad.
Condor convierte a una colección de estaciones de trabajo en un entorno de computación distribuida de alto disponibilidad (HTC).
Como la mayoría de los sistemas por lotes, Condor brinda mecanismos para encolamiento de trabajos, políticas de planificación, esquema de prioridades y clasificación de recursos.
Los usuarios envían su trabajo a Condor, este los encola, posteriormente los ejecuta e informa los resultados al usuario [2].

4.1 Acceso al pool

En la arquitectura disponible para ejecutar trabajos bajo Condor (ver Figura 1), el administrador central que recoge toda la información del pool es Glapic y tiene derechos de administrativos sobre el mismo actuando como negociador entre los recursos y las solicitudes de recursos. Eventualmente se podría acceder directamente a Reloaded, pero el esquema actual permitirá manejar desde Glapic otros recursos además del cluster.
Estas tareas las ejecutan distintos demonios. Los demonios de Condor se ejecutados bajo el usuario root.
Por otra parte los usuarios se conectan al administrador central con el usuario condor, y desde esta máquina central se emiten los trabajos al pool.
Esta máquina cumple un rol muy importante en el pool, debe ser confiable y está en funcionamiento permanente. Si este equipo deja de trabajar, no podrán realizarse asociaciones entre recursos y requerimientos de recursos.

4.2 Autenticación y autorización de usuarios

La autenticación de los usuarios en la máquina que emite los trabajos, se basa en políticas estándares de cuentas de sistemas operativos.
Cuando un usuario quiere acceder al pool para procesar trabajos, Condor utiliza un identificador de usuario y de grupo (user ID, group ID) para determinar quién es ese usuario. Para el usuario condor que se ha creado, el identificador de usuario y de grupo es configurado a 2.2

Una vez que el usuario es autenticado, Condor necesita conocer los privilegios que tiene ese usuario. Para ello, en el archivo de configuración general de Condor llamado `condor_config`, se fija el valor de la variable `HOSTALLOW_READ=*`, indicando que todos pueden ver el estado del pool, pero no pueden unirse o ejecutar trabajos en el mismo.
Luego se configura la variable `HOSTALLOW_WRITE` para que únicamente las máquinas dentro del dominio de interés puedan emitir trabajos y unirse al pool.
Los administradores centrales realizan la asociación de los recursos con trabajos y recogen información de todo el pool. Las máquinas del pool anuncian sus características en avisos clasificados (ClassAds), así como también los trabajos anuncian sus requerimientos y preferencias. Un ejemplo de esto es presentado en la Tabla 1

Tabla 1. Ejemplo de ClassAdd

```
NameOpSysArch State Activity LoadAv Mem ActvtyTime
vm1@compute-0 LINUX INTELOwner Idle 1.000 504 0+00:55:08
vm2@compute-0LINUX INTEL Owner Idle 2.000 504 0+00:19:34
vm1@compute-0LINUXINTELUnclaimedIdle0.0005040+00:15:04
vm2@compute-0LINUXINTELUnclaimedIdle0.0005040+00:18:18
vm1@compute-0LINUX INTELOwner Idle1.000 504 0+00:50:07
vm2@compute-0LINUX INTEL Owner Idle 2.000 504 0+00:15:11
vm1@compute-0LINUX INTEL Owner Idle 1.000 504 0+00:50:13
vm2@compute-0LINUX INTEL Owner Idle 2.060 504 0+00:20:05
vm1@compute-0LINUX INTEL Owner Idle 1.000 504 0+00:45:34
vm2@compute-0LINUX INTEL Owner Idle 2.000 504 0+00:35:31
vm1@compute-0LINUX INTEL Owner Idle 1.000 504 0+00:45:10
vm2@compute-0LINUX INTEL Owner Idle 2.000 504 0+00:20:39
vm1@reloadedLINUXINTELUnclaimedIdle 0.010 504 0+00:17:30
vm2@reloadedLINUXINTELUnclaimedIdle 0.000 504 0+00:17:29
```

El demonio `condor_negotiator` es el responsable de las asociaciones entre recursos y solicitudes de recursos dentro del pool. Este demonio inicia periódicamente un ciclo de negociación, consultando a la máquina que actúa como recolector de información del pool (es decir el collector), sobre el estado de los recursos.
El demonio `condor_schedd` representa las solicitudes de recursos dentro del pool. Los demonios `condor_collector` y `condor_negotiator` se ejecutan en el administrador central.
El demonio `condor_collector` es el responsable de recoger información sobre el estado del pool.

4.3 Ejecución de trabajos

Para procesar un programa en Condor, se debe ejecutar un trabajo por lotes, en background. Con este fin es necesario crear los archivos que contienen las entradas correspondientes para el programa a ejecutar.
En el archivo de emisión del trabajo (ver Figura 2), se especifica si Condor debe transferir los archivos necesarios para la ejecución del trabajo.
En la Figura 2, `Requirements` indica requerimientos específicos y `Rank` indica preferencias, para que dentro de los recursos que cumplan los requerimientos, el trabajo se ejecute en máquinas determinadas.
Para posibilitar el mecanismo de transferencia, se colocan dos instrucciones en el archivo de emisión del trabajo: `should_transfer_files`, la cual especifica que Condor debe transferir archivos desde la máquina que emite el trabajo a la máquina remota donde se ejecuta. y `when_to_transfer_output` que especifica cuándo transferir los resultados [4].

Así, Condor transfiere los archivos al nodo, lo ejecuta y transfiere la salida a la máquina que emitió el trabajo.

```
universe=vanilla
executable = gaids.run
Rank= (machine == "compute-0-6.local") || (machine == "compute-0-
5.local)
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
transfer_output_files = results.output
output=results.output
error=error.output
log=results.log
queue 30
```

Figura 2. Archivo de emisión de un trabajo

Se puede saber dónde se está ejecutando el trabajo con el comando `condor_q`, de acuerdo a lo especificado en la condición `Rank` del archivo de emisión del trabajo. Un ejemplo de la salida de `condor_q` se muestra en la Tabla 2

Tabla 2. Salida del comando condor_q

```
ID OWNER SUBMITTED RUN\_TIME HOST(S)
28.0 condor 6/8 10:49 0+00:27:54 vm2@-0-1.local
28.1 condor 6/8 10:49 0+00:27:57 vm2@0-8.local
28.2 condor 6/8 10:49 0+00:33:05 vm1@reloaded
28.3 condor 6/8 10:49 0+00:33:02 vm2@reloaded
28.4 condor 6/8 10:49 0+00:27:54 vm2@0-5.local
28.5 condor 6/8 10:49 0+00:18:08 vm2@0-3.local
28.6 condor 6/8 10:49 0+00:18:06 vm2@0-2.local
28.7 condor 6/8 10:49 0+00:07:42 vm2@0-9.local
28.8 condor 6/8 10:49 0+00:06:17 vm2@0-1.local
28.9 condor 6/8 10:49 0+00:06:15 vm1@reloaded
28.10 condor 6/8 10:49 0+00:06:12 vm2@reloaded
28.11 condor 6/8 10:49 0+00:02:48 vm2@0-9.local
```

En el archivo de emisión de un trabajo se puede también especificar la un archivo para llevar un registro de lo que sucede con el trabajo.
Cuando el trabajo termina, los resultados se envían a la máquina que lo emitió. Por lo tanto, si hay muchos trabajos ejecutándose, se necesita una gran cantidad de memoria real y de espacio de intercambio en el administrador central [5].

4.4 Checkpointing

El mecanismo de checkpointing permite obtener el estado actual de un programa a fin de que pueda ser reiniciado posteriormente desde ese último estado.
Condor a través del planificador puede quitar un recurso a un trabajo, efectuar el checkpointing del trabajo y reiniciarlo más tarde sin perder lo hecho hasta ese punto. Por defecto, se genera un archivo de checkpoint que se almacena en el disco local de la máquina que emitió el trabajo.

5 ENTORNOS DE EJECUCIÓN EN CONDOR

Condor ofrece varios entornos de ejecución (llamados universos) de los cuales podemos elegir a la hora de ejecutar un trabajo, y este universo elegido se especifica en el archivo de emisión del trabajo (ver Figura 2). Los distintos universos disponibles en Condor se comentan seguidamente:

Standard: permite checkpointing y ejecución remota de llamadas al sistema. Estas características hacen que un trabajo sea más confiable y le permite acceder de manera uniforme a todos los recursos del pool. Para que un trabajo se pueda procesar en el universo standard debe enlazarse con el comando `condor_compile`. Condor efectúa checkpointing de los trabajos a intervalos regulares. Si un trabajo debe migrarse de una máquina a otra, a través del mecanismo de checkpointing, Condor genera una imagen del estado del trabajo, copia la imagen en la otra máquina y reinicia el trabajo desde el punto en que se dejó de ejecutar. Los trabajos pueden ejecutarse en diferentes máquinas a lo largo de su vida, gracias a la ejecución de remota de llamadas al sistema.

Vanilla: este universo permite ejecutar trabajos que no pueden ser relinkeados contra Condor. En este universo es posible realizar checkpointing y ejecución remota de llamadas al sistema. Si un trabajo es terminado parcialmente, Condor puede suspenderlo para completarlo posteriormente o reiniciarlo en otra máquina del pool. Como no puede efectuarse la ejecución remota de llamadas al sistema, Condor trabaja con un sistema de archivos compartidos, como NFS o AFS. Alternativamente, Condor puede transferir los archivos que necesita el trabajo para ejecutarse.

PVM: permite la ejecución de programas escritos para Parallel Virtual Machine

MPI: permite la ejecución de programas escritos para MPICH.

Grid: brinda a los usuarios una interfaz Condor estándar para que puedan ejecutar trabajos en sistemas de administración remotos.

Java: un programa especificado para el universo Java puede ejecutarse en cualquier máquina con JVM, más allá de su ubicación, propietario o versión de JVM.

Scheduler: permite a los usuarios la ejecución de trabajos livianos en la misma máquina que los emitió. Este universo tiene posibilidades muy limitadas, por ello cuando el trabajo debe ejecutarse en la máquina que lo envía, se prefiere el universo local.

Paralelo: permite la ejecución de programas paralelos, por ejemplo los trabajos MPI.

Local: en este universo, el trabajo no espera asociarse con ningún recurso, ya que se ejecuta directamente en la máquina que lo emitió.

Cuando se prepara el archivo para emitir el trabajo a ejecutar se debe brindar información acerca del mismo, como cuál es el archivo ejecutable, los archivos que se necesitan como entrada, la plataforma necesaria para ejecutar el trabajo, la cantidad de veces que debe ejecutarse, etc.
Los agentes y los recursos anuncian sus características y requerimientos en anuncios clasificados (ClassAds). Condor, realiza asociaciones dando especial importancia a dos atributos especiales del archivo de emisión del trabajo: `Requirements` y `Rank`. `Requirements` indica requerimientos específicos y `Rank` indica preferencias, para que dentro de los recursos que cumplan los requerimientos, el trabajo se ejecute en máquinas determinadas.
El algoritmo de asociación necesita que ambos ClassAds coincidan y que los `Requirements` correspondientes sean evaluados como verdaderos.

6 RECONOCIMIENTO DE PATRONES EN EL TRÁFICO DE RED

6.1 Introducción al problema

Las falencias en la seguridad de protocolos como ARP, TCP, TELNET, SMTP, FTP han sido la causa de ataques contra la confidencialidad, la disponibilidad y la autenticidad de los datos transportados. Si bien estos problemas han sido corregidos a lo largo de los años, continuamente se van descubriendo nuevas maneras de realizar estos ataques.
El ingeniero en seguridad de redes debe estar alerta para detectar estos ataques, informándose de las nuevas vulnerabilidades descubiertas o tipos de ataques perpetrados. Sin embargo una gran cantidad de estos ataques tienen lugar antes que se conozcan siquiera las vulnerabilidades o fallas
que los provoca.
Para hacer frente a esto es que en los últimos años han surgido propuestas para la aplicación de técnicas de inteligencia artificial en el ámbito de la seguridad en redes [6,7,8].
Con este objetivo se utiliza un algoritmo genético para el reconocimiento de patrones en el tráfico de red como punto de partida para abordar un problema de mayor envergadura como lo es la detección de intrusos por anomalías en el tráfico de red.
Entendiéndose por una anomalía a toda instancia de tráfico que se aparte del comportamiento normal de la red [9].

6.2 Algoritmo genético propuesto

El algoritmo genético propuesto parte de una población de individuos conformados por instancias de tráfico de red elegidas al azar, para al finalizar obtener el conjunto de reglas que más coincidencias encuentre en el tráfico de la red.
Para la representación de la población se seleccionan 6 atributos de una instancia de tráfico: tiempo de duración de la conexión, tipo de protocolo, puerto origen, puerto destino, dirección IP origen y dirección IP destino.
Se propone una función de fitness definida en la ecuación (1) que favorece a aquellos individuos de la población que presenten mayor cantidad de coincidencias en los atributos de las instancias de tráfico.

$$f(r) = \frac{\prod_{j=1}^{m} \prod_{i=1}^{n} \alpha(r_i, d_{ji})}{|D|} \quad (1)$$

Para calcular el valor de fitness del individuo *r*, se comparan los genes del individuo *r* con los correspondientes d_{ji} de cada una de las instancias de tráfico pertenecientes al conjunto de entrenamiento *D* mediante la función ? definida como:

$$\alpha(r_j, d_{ji}) = \begin{array}{c} w_i \text{ si } r_j = d_{ji} \\ w'_i \text{ si } r_j \neq d_{ji} \end{array} \quad (2)$$

Donde cada gen tiene un peso asociado w_i con el objeto de favorecer a aquellos individuos que por experiencia disciplinar resultan más relevantes. Cuando algún gen no presente coincidencia, se le asigna un peso w'_{ij}, que se ajusta en la práctica al valor 0,1.
De esta manera la función de fitness favorece a los genes que presentan una frecuencia de aparición relativamente alta. Estos individuos en futuras generaciones, pueden originar nuevas reglas que coincidan con un significativo número de instancias de tráfico.

Con el fin de comprobar el funcionamiento del algoritmo propuesto se realizan experimentos que utilizan 2 conjuntos conformados con instancias de tráfico tomadas del conjunto de datos provisto por DARPA [10]. El primer conjunto contiene 9000 entradas que representan 4 horas de tráfico, utilizado para la etapa de entrenamiento. El segundo conjunto contiene 35000 instancias de tráfico que representan 24 horas, utilizado para la fase de prueba.
El algoritmo necesita de un ajuste en sus parámetros con el fin de encontrar la combinación que presente mejores resultados.
Sin embargo debido a la característica no determinística de los algoritmos genéticos, este necesita ser ejecutado un determinado número de veces para luego mediante pruebas estadísticas determinar si alguna combinación de parámetros constituye efectivamente una mejora significativa.
Debido al alto consumo de recursos computacionales de la función de fitness de cada ejecución, el ajuste de parámetros de un algoritmo genético resulta una tarea que puede demandar mucho tiempo. Si embargo al tratarse de una tarea inherentemente paralela, se puede pensar en la utilización de computación distribuida.

6.3 Implementación computacional

EL algoritmo fue implementado en Python [11] por tratarse de un lenguaje de alto nivel que permite desarrollar prototipos funcionales en poco tiempo. Al existir versiones de la maquina virtual de Python para múltiples sistemas operativos, esto permite la portabilidad.
La aplicación esta compuesta por una serie de archivos que conforman los distintos módulos de la aplicación.
Al tratarse de un lenguaje interpretado que se ejecuta sobre una maquina virtual los tiempos de ejecución resultan mucho mayores a los de un programa que se ejecuta de manera nativa. Para solucionar este problema se utilizo un compilador en tiempo de ejecución [12] (pysco)

6.4 Adaptación de la aplicación al entorno Condor

Se comentan a continuación los pasos necesarios para la adaptación de la aplicación de reconocimiento de patrones de tráfico al entorno Condor. Cabe destacar que al no contar con un universo Python dentro del entorno Condor, se debe utilizar el universo vanilla, el cual no provee muchas de las principales funcionalidades de Condor.
Para poder ejecutar el código del algoritmo genético en el entorno Condor es necesario satisfacer los siguientes requisitos:

- El código puede ser ejecutado en diferentes sistemas operativos y arquitecturas de hardware.
- El código debe poder ser fácilmente transferido al recurso remoto.

El primer requisito se satisface fácilmente, ya que en encuentran implementaciones del interprete de Python para la mayoría de los sistemas operativos utilizados en la actualidad.
Con el objetivo de transferir los datos y el código de la aplicación, se utiliza un archivo auto descomprimible que contiene toda la información necesaria para la ejecución de la aplicación en el recurso remoto.
La aplicación es empaquetada en un único archivo usando la herramienta cx_freeze [13]. Este archivo a su vez es enlazado contra una biblioteca estática que contiene la maquina virtual de Python. Al final de este proceso se obtiene un ejecutable ubicado en un directorio previamente definido por el usuario.
El lenguaje Python cuenta con un número de módulos dinámicos que no pueden ser enlazados de manera estática al ejecutable previamente creado por cx_freeze. Por lo que, en un segundo paso,

estos módulos como así también los archivos de datos requeridos por el programa, son copiados al directorio definido por el usuario. Finalmente el archivo ejecutable, los archivos de datos y los módulos dinámicos de python son incluidos en un archivo auto descomprimible utilizando la herramienta makeself.sh [14]. De esta manera se obtiene un único archivo que puede ser fácilmente transferido a un recurso remoto.
Esta herramienta permite incluir en el archivo auto descomprimible un script que contiene todos los comandos necesarios para realizar la ejecución remota de la aplicación. Se incluye además las instrucciones necesarias para la creación de los archivos de salida basados en el nombre del recurso remoto y una marca de tiempo. De esta manera se evitan posibles colisiones debido a las múltiples ejecuciones de la aplicación en un mismo nodo.

6.5 Resultados obtenidos

El tiempo total de ejecución de la aplicación serial en un P4 HT con 1GB de RAM es de aproximadamente 15 minutos, para validar estadísticamente cada ajuste realizado al algoritmo es necesario ejecutar el programa un mínimo de 30 veces por lo que el tiempo total de ejecución en la misma maquina serial es cercano a las 7 horas.
En el trabajo serial [3] se presentaron los resultados obtenidos al ejecutar algoritmo durante 1200 generaciones. La elección del número de generaciones se debe principalmente a las limitaciones impuestas por los requerimientos computacionales de la función de fitness.
Los resultados obtenidos se presentan en la Figura 3. Al finalizar le ejecución del algoritmo se obtuvieron reglas con atributos que permiten encontrar coincidencias sobre aproximadamente el 85% del conjunto de entrenamiento y un 80% sobre el conjunto de prueba. El histograma de la Figura 3 muestra que solamente el 25% de las ejecuciones de la aplicación obtienen un porcentaje de error de clasificación cercano al 15% en el conjunto de entrenamiento e incluso se obtienen resultados cercanos al 56% de error de clasificación

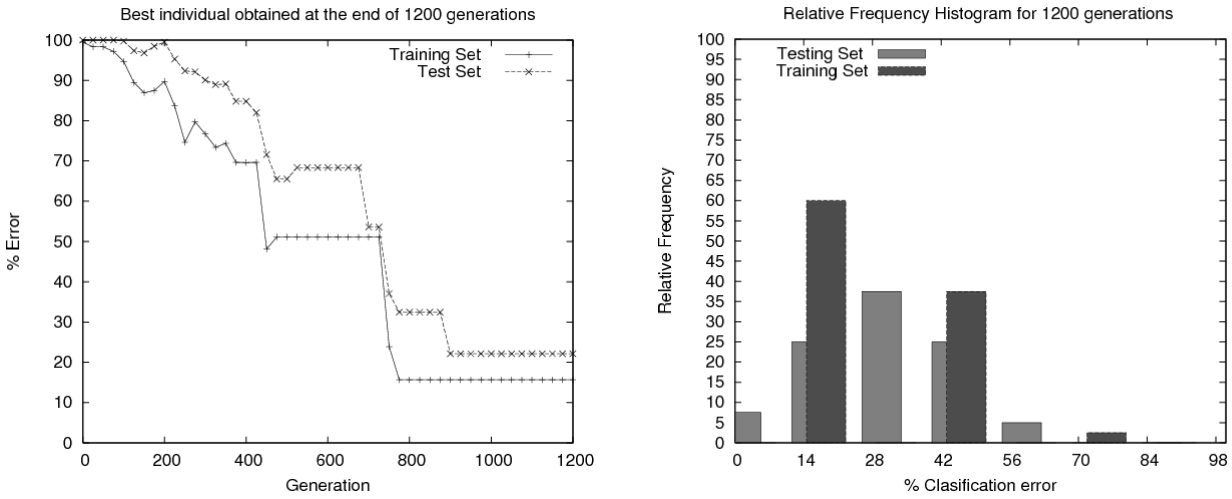


Figura 3. Resultados del mejor individuo obtenido y histograma de frecuencias relativas de la aplicación serial.

La posibilidad de ejecutar la aplicación del algoritmo genético en un entorno Condor ha permitido extender el número de generaciones del algoritmo a 1800. En este caso se utilizan todos los recursos computaciones disponibles de la Figura 1. Cabe destacar que son necesarios tres ciclos de ejecución para ejecutar las 30 corridas del algoritmo genético propuesto, debido a que solo se cuenta con la disponibilidad de 14 maquinas al mismo tiempo. En este caso el tiempo total de ejecución es de 67

minutos. Este resultado se debe a que el tiempo total de ejecución del algoritmo a lo largo de 1800 es de 20 minutos.

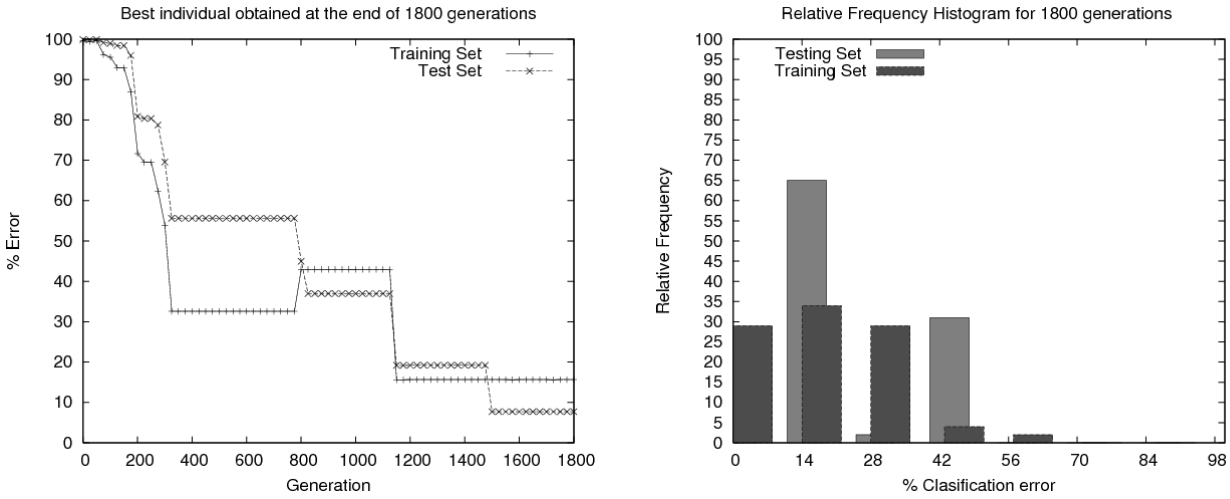


Figura 4. Resultados del mejor individuo obtenido y histograma de frecuencias relativas de la aplicación ejecutada en entorno Condor

Las pruebas preliminares del algoritmo en un entorno Condor han permitido mejorar los resultados anteriores. Luego de 1800 generaciones se obtuvo un conjunto de reglas con atributos que permiten encontrar coincidencias sobre aproximadamente el 85% del conjunto de entrenamiento y un 93% sobre el conjunto de prueba. La frecuencia de aparición de buenas soluciones también presentó una considerable mejora ya que nunca se supera el 42% de error de clasificación en el conjunto de prueba y un 65% de las ejecuciones presentan un error de clasificación cercano al 15%

7 CONCLUSIONES

En este documento se ha discutido la ejecución de una aplicación para la búsqueda de patrones de tráfico de red en un entorno Condor, con el objetivo de ser utilizados en Sistemas de Detección de Intrusos. La ejecución de esta aplicación ha permitido comprobar las capacidades de Condor a la hora de planificar y monitorear los trabajos y los recursos disponibles.

Condor brinda al usuario la posibilidad de ejecutar trabajos con requerimientos elevados, inclusive si los recursos no están temporalmente disponibles, ofreciendo capacidades únicas como migración de procesos y checkpointing.

Estas facilidades otorgan un alto poder computacional haciendo posible que cálculos con alto costo de gestión puedan ser realizados en forma ágil y eficiente.

Con una infraestructura de 12 máquinas bajo un entorno Condor, se obtiene una reducción del tiempo total de ejecución requerido por la aplicación. Por otra parte el poder computacional ofrecido por Condor ha permitido extender el número de generaciones consideradas en el algoritmo genético, obteniendo reglas de mejor calidad para el reconocimiento de patrones en el tráfico de red. Cabe destacar que la aplicación serie original se mantuvo sin cambios.

En un futuro, se planea configurar varios pool y administrarlos con Condor, así como también administrar máquinas con doble microprocesador, que permite dos instancias de trabajo a la vez.

REFERENCIAS

[1] Sanz A. Condor. Manual de usuario para el cluster HERMES, Instituto de Investigación en ingeniería de Aragón, Universidad de Zaragoza. Disponible en: http://i3a.unizar.es/hermes/manual_hermes.pdf, 2006.

[2] Thain D., Tannenbaum T., and Livny M., Distributed Computing in Practice: The Condor Experience, Computer Sciences Department, University of Wisconsin-Madison. Disponible en: http://www.cs.wisc.edu/condor/publications.html, 2002.

[3] Catania C. y Garcia Garino C. Una propuesta de reconocimiento de patrones en el tráfico de red basada en algoritmos genéticos the 9th Argentinian Symposium on Artificial Intelligence, ASAI, 2007.

[4] Livny M., Basney J., Raman R. y Tannenbaum T. Mechanisms for High Throughput Computing, Department of Computing Sciences, University of Wisconsin-Madison. Disponible en: http://www.cs.wisc.edu/condor/publications.html, 1997.

[5] Condor Team, Condor® Version 6.8.1 Manual, University of Wisconsin–Madison, Disponible en: http://www.cs.wisc.edu/condor/manual/, 2006.

[6] Bridges S. and Vaughn R. Fuzzy data mining and genetic algorithms applied to intrusion deteccion. In National Information Systems Security Conference, 2000.

[7] Gong R., Zulkernine M., and Abolmaesmumi P. A software implementation of a genetic algorithm based approach to network intrusion detection. In Sixth SNPD/SAWN, 2005.

[8] Li W. and Traore I. Detecting new forms of network intrusion using genetic programming. Computational Intelligence, Vol 20, 475-494, 2004.

[9] Mukherjee B., Heberline L. T. and Levitt K. Network instruction detection. IEEE Network, 1994.

[10] Mit lincoln laboratory DARPA data set. Disponible en http://www.ll.mit.edu/IST/ideval/data/dataindex.html.

[11] Python Programming Language. Disponible en http://www.python.org.

[12] Rigo R. Psyco, the python specializing compiler. Disponible en: http://psyco.sourceforge.net/psyco.ps.gz, 2001.

[13] C. C. Ltd. cx freeze, a set of utilities for freezing python scripts into executables. Disponible en: http://python.net/crew/atuining/cx_Freeze/.

[14] Peter S. makeself, make self-extractable archives on unix. Disponible en: http://www.megastep.org/makeself/.

Superlinealidad sobre Clusters. Análisis experimental en el problema del Puzzle N^2 -1.

Victoria Sanz [1], Franco Chichizola [2], Marcelo Naiouf [3], Laura De Giusti [4], Armando De Giusti [5]

Instituto de Investigación en Informática (III-LIDI) – Facultad de Informática – UNLP

Abstract

This paper analyzes the scalability of superlinear parallel algorithms run over cluster architectures.
The case study is the resolution of the N^2 -1 Puzzle problem and, in it, the implementation of a parallel solution over clusters -based on the A* algorithm- is analyzed.
In particular, the results relating the speedup in function of the number of processors in the clusters are presented together with a discussion on the superlinearity and efficiency (or pseudo-efficiency) attainable when scaling the architecture and / or the dimension of the problem.
Finally, some future research lines, oriented to predicting the attainable superlinearity in function of the initial disorder degree, are presented.

Keywords: *Parallel Algorithms, Superlinearity, Scalability, Efficiency. Cluster Architectures.*

Resumen

En este trabajo se analiza la escalabilidad de algoritmos paralelos superlineales ejecutándose sobre arquitecturas de cluster.
El caso de estudio es la resolución del problema del Puzzle N^2 -1 y en él se analiza la implementación de una solución paralela sobre clusters, basada en el algoritmo A*.
En particular se presentan resultados que relacionan el speedup en función del número de procesadores en el cluster y se discute la superlinealidad y eficiencia (o seudoeficiencia) alcanzable al escalar la arquitectura y/o la dimensión del problema.
Por último se presentan líneas de investigación futuras orientadas a predecir la superlinealidad alcanzable en función del grado de desorden inicial.

Palabras Clave: *Algoritmos paralelos. Superlinealidad. Escalabilidad. Eficiencia. Arquitecturas de Cluster.*

VI Workshop de Procesamiento Distribuido y Paralelo.

[1] Becaria Alumna. Auxiliar Docente de la Facultad de Informática UNLP. vsanz@ciudad.com.ar.
[2] Becario de Doctorado del CONICET. Profesor Adjunto de la Facultad de Informática UNLP. francoch@lidi.info.unlp.edu.ar.
[3] Profesor Titular D.E. Facultad de Informática UNLP. mnaiouf@lidi.info.unlp.edu.ar.
[4] Profesor Adjunto de la Facultad de Informática UNLP. ldgiusti@lidi.info.unlp.edu.ar
[5] Investigador Principal CONICET. Profesor Titular. Facultad de Informática UNLP. degiusti@lidi.info.unlp.edu.ar.

[*] *Esta investigación es parcialmente financiada por la CIC y, la Fundación YPF.*

1. INTRODUCCIÓN

El primer punto de interés en la resolución de un algoritmo sobre una arquitectura multiprocesador es el *factor de Speedup* ***Sp*** que es una medida de perfomance relativa, definida como la relación entre el tiempo de ejecución del mejor algoritmo secuencial sobre una máquina monoprocesador y el tiempo de ejecución del algoritmo paralelo correspondiente sobre una máquina multiprocesador [1] [2]. Si llamamos Ts al tiempo de ejecución secuencial y Tp al paralelo, tenemos la relación ***Sp = Ts/Tp*** que normalmente se trata de maximizar en el desarrollo de aplicaciones paralelas.

Normalmente el Speedup está limitado por el máximo grado de concurrencia que puede obtenerse de la aplicación, por el inevitable componente secuencial del algoritmo y por el número de procesadores ***N*** disponibles para la ejecución. Dos conceptos surgen para la mayoría de los algorimos paralelos:

- El incremento del número de procesadores utilizados puede hacer crecer el Speedup, pero este crecimiento está acotado por el máximo grado de concurrencia alcanzable. A partir de este punto agregar procesadores no incrementará el Speedup, e incluso puede disminuirlo [3].
- El máximo teórico del Speedup está dado por el número de procesadores, es decir ***Sp <= N*** [4].

Un segundo parámetro de importancia al analizar aplicaciones paralelas es la *Eficiencia* ***E*** alcanzada. Se define como Eficiencia la relación entre el Speedup y el número de procesadores utilizados para obtenerlo: ***E = Sp / N.*** Esta definición pone la Eficiencia entre 0 y 1. Alcanzar valores cercanos a 1 está significando que se logra Speedup cercano al óptimo N.

La Eficiencia resulta una métrica de calidad y de costo del algoritmo paralelo que es particularmente importante y no siempre se puede mantener al escalar los problemas, al incrementar el número de procesadores o al portar el algoritmo sobre otra arquitectura multiprocesador [3] [5]. Mantener la eficiencia constante al escalar un algoritmo paralelo se denomina isoeficiencia y en muchas aplicaciones es un objetivo lograr algoritmos que tengan esta propiedad [1].

La *Escalabilidad* es un factor muy importante en las aplicaciones paralelas: normalmente los problemas "escalan" es decir aumenta el volumen de trabajo a realizar y también las arquitecturas multiprocesador que utilizamos pueden "escalar" incrementando los procesadores utilizados. Es de interés investigar el efecto de escalar trabajo y/o procesadores sobre el rendimiento de los algoritmos paralelos, considerando Speedup y Eficiencia [2] [3].

El máximo Speedup teórico puede en algunos casos ser mejorado y esto da lugar al concepto de *Superlinealidad* ***Su.*** Es interesante analizar por qué el Speedup puede superar N:

- En la mayoría de los casos se han utilizado algoritmos secuenciales sub-óptimos (al crecer Ts crece Sp) y la relación de cálculo de Sp no respeta la definición que indica emplear el mejor algoritmo secuencial. Estos casos podemos considerarlos como una "falla".
- En otras oportunidades, la memoria disponible en la arquitectura monoprocesador es mucho menor que la memoria total utilizable en la arquitectura paralela, con lo que un esquema de paralelización por división del espacio de datos puede conducir a un Speedup superlineal al no requerir accesos a memoria secundaria [6].
- La clase de aplicaciones que nos interesan y pueden dar superlinealidad al paralelizarlos son los problemas de optimización discreta, donde la exploración del espacio total de soluciones posibles puede reducirse al distribuir el trabajo entre N procesadores y poder "cortar" o "finalizar" la búsqueda global al llegar al resultado esperado en cualquiera de ellos [7] [8].

Los problemas de optimización discreta abarcan un gran número de áreas [9] y a menudo son resueltos con métodos de exploración del espacio de estados, buscando un estado "solución" [10]. Estas técnicas de búsqueda normalmente tienen un alto costo computacional y permiten una paralelización natural, con lo que los algoritmos de resolución paralela en optimización discreta han sido motivo de numerosos desarrollos y publicaciones dentro del procesamiento paralelo [11] [12].

En particular las técnicas de BFS (Best First Search) parten de nodo o estado inicial del grafo que represente el problema que es seleccionado por alguna métrica de estimación del trabajo para llegar a la solución y tratan de evolucionar a partir de este punto. La paralelización natural de la técnica consiste en iniciar la evolución desde diferentes nodos "posibles" sobre los distintos procesadores de la arquitectura multiprocesador. A medida que el algoritmo evoluciona es necesario comunicar los procesadores para informar resultados alcanzados o bien soluciones descartadas de acuerdo a la métrica elegida [13] [14].

Es interesante reflexionar sobre algunos aspectos que se dan al utilizar arquitecturas paralelas tipo cluster en la resolución de problemas de optimización discreta [15]:

- La granularidad de la paralelización es crítica, porque de ella dependerá la mejora en el tiempo de solución y también el overhead de comunicaciones.
- En general el balance de carga tiene que ser dinámico (lo que exige comunicación) ya que el trabajo exploratorio es variable y es muy difícil predecirlo a priori [16].
- Conceptualmente podría esperarse que para una dimensión fija del problema a resolver (espacio total de nodos a explorar), el incremento del número de procesadores permita escalar el Speedup hasta un punto, donde las comunicaciones limiten el crecimiento.
- En teoría la arquitectura de cluster podrá permitirnos alcanzar superlinealidad, dependiendo del balance de carga, la heterogeneidad de los procesadores y la relación tiempo de procesamiento/tiempo de comunicaciones del algoritmo empleado [17].

Caso de Estudio y Contribución

En este trabajo se estudia el caso del Puzzle N^2 -1 [18][19], desarrollando una solución paralela con una técnica de BFS sobre un cluster.
En particular se analiza la superlinealidad alcanzable para distintas dimensiones del problema y diferente número de procesadores.
Asimismo se analiza la máxima eficiencia alcanzable y el número de procesadores en el cluster para alcanzar un nivel de eficiencia dado.

2. ANALISIS DEL PROBLEMA DEL PUZZLE N^2 -1

El problema del Puzzle N^2-1 es una generalización del Puzzle-15 ideado por Sam Lloyd [20][21]. Consiste en N^2-1 piezas numeradas de 1 a N^2-1 colocadas en un tablero de tamaño N^2.

N^2-1 casilleros del tablero contienen exactamente una pieza, quedando sólo una casilla vacía la cual se denomina "hueco".

El objetivo del puzzle es repetidamente llenar el hueco con una pieza adyacente a él en sentido horizontal o vertical, hasta alcanzar un tablero donde en la casilla (i,j) se encuentra la pieza numerada como (i-1)*N + j y en la casilla (N,N) se encuentra el hueco.

La solución al problema planteado tendrá que ser aquella que minimice la cantidad de movimientos que deben realizarse para alcanzar la configuración final desde la configuración inicial dada.

La Figura 1.a muestra un Puzzle N^2-1 donde N es 4. La Figura 1.b representa un esquema de solución al problema.

2	5	1	12
15	3	8	14
4	13	10	
11	6	9	7

Figura 1.a. Tablero inicial

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Figura 1.b. Tablero final.

Construcción del espacio de estados del problema

Para construir el grafo sobre el cual se realizará la búsqueda se requiere:

- *Esquema de ramificación:* utilizado para generar subproblemas a partir de un problema dado.
- *Estrategia de búsqueda:* indica cómo seleccionar un nodo entre los nodos pendientes de acuerdo a prioridades definidas. Generalmente la estrategia se basa en seleccionar nodos que estén próximos a la solución.
- *Función de costo:* estima el costo de alcanzar la solución a partir de un nodo intermedio. Sea h(x) esta función (también llamada *heurística*); si h(x) es una cota inferior del costo real requerido para alcanzar el nodo solución a partir de x, entonces h(x) es admisible[6]. Sea g(x) la función que denota el costo de alcanzar el estado x a partir de la configuración inicial, luego la cota inferior para el camino que puede ser obtenido expandiendo el camino actual entre la configuración inicial y el estado x es L(x) = g(x) + h(x).

Distancia de Manhattan

Supongamos que cada posición del tablero se representa como un par ordenado. La distancia entre las posiciones (i,j) y (k,l) está definida como | i - k | + | j - l |. Dicha distancia recibe el nombre de *Distancia de Manhattan*. La suma de las distancias de Manhattan entre las posiciones del tablero x y el tablero final será un estimador mínimo del número de movimientos requeridos para transformar el tablero x en el tablero solución. Esta función se utilizará como función heurística admisible h(x).

Algoritmo A*

A* es una de las variantes más conocidas de la búsqueda Best First Search. Cada nodo n es valuado de acuerdo con el costo de alcanzar el mismo a partir de la raíz del árbol de búsqueda (g(n)) y una heurística que estima el costo para ir de n hasta un nodo solución (h(n)). Luego, la función de costo será L(x) = g(x) + h(x).
Está garantizado que el algoritmo A* siempre encontrará la mejor solución.

[6] Función admisible h(x): el costo real para alcanzar el nodo objetivo a partir del nodo x siempre es mayor o igual al valor de h(x).

A* secuencial

Este algoritmo mantiene una lista de nodos no explorados (lista abierta) ordenada de acuerdo al valor de la función L, y otra lista de nodos ya explorados (lista cerrada). Inicialmente la lista abierta contiene un solo elemento, el nodo inicial, y la lista cerrada esta vacía.

En cada paso, el nodo con menor valor L (el *mejor nodo*) es removido de la lista abierta y es examinado. Si es el nodo solución, el algoritmo termina. Caso contrario, el nodo es expandido y es insertado en la lista cerrada. Cada nodo sucesor es insertado en la lista abierta sólo si no estaba en la lista cerrada, o estaba pero su valor L es menor al anterior.
La lista cerrada es utilizada para evitar trabajo repetido, de modo que no se expandirán nuevamente nodos ya explorados.

Dado que siempre se remueve el nodo con menor valor L, la lista abierta comúnmente se implementa con una cola de prioridades. La lista cerrada se puede implementar con una tabla de hash , donde cada entrada contiene como clave un valor de la heurística h(x), y como valor una lista con todos los tableros cuyo h(x) es igual a la clave.

Si se requiere conocer el camino desde el nodo inicial hasta el nodo solución, cada nodo debe mantener una referencia a su nodo padre, o poseer la secuencia de pasos por la cual se llegó hasta él, así se podrá determinar cómo fue encontrado.

El esquema secuencial es el siguiente:

```
Crear lista abierta.
Crear lista cerrada.
Insertar (lista abierta, x, h(x)).
mientras (lista abierta no vacía) y (no encontré solución)

// Extraer el mínimo nodo de la lista abierta, llamémoslo n
n = EliminarMínimo (lista abierta)

// Si el nodo era la solución, es decir h(n) = 0, termina el algoritmo.
// Caso contrario se expande n.
si (EsSolución(n))
        solución = n
sino
        hijos = Expandir(n)
        Insertar(lista cerrada,n)

// Un nodo es aceptable si no esta en la lista cerrada, o si está pero con un costo mayor al actual
        para cada uno de los hijos de n
                si (EsAceptable(hijo))
                        Insertar(lista abierta, hijo, h(hijo) + g(n) + 1)
Retornar solución.
```

A* paralelo

La estrategia de paralelización consiste en mantener las listas *abierta* y *cerrada* locales para cada procesador. Al principio sólo uno de los procesadores tendrá como trabajo el nodo inicial. A medida que son generados otros nodos, los procesadores recibirán los mismos para comenzar a trabajar.

Todos los procesadores harán una búsqueda en el ámbito local, construyendo su propia lista cerrada, para evitar trabajo repetido localmente, como también su lista abierta local. Los procesadores deben comunicarse los valores mínimos de las soluciones encontradas, a fin de minimizar búsquedas innecesarias.

La solución paralela planteada no requiere un proceso central, sólo requiere procesos trabajadores que realicen la búsqueda en el espacio de estados del problema. Para la implementación del algoritmo paralelo se utiliza la técnica de balance de carga distribuida "Asynchronous Round Robin" (ARR) y el algoritmo de "Terminación de Dijkstra" modificado [22].

Dados p procesos trabajadores, cada uno dispondrá de su lista abierta y lista cerrada de nodos, así como un valor indicando el costo de la mejor solución encontrada hasta el momento (CMS), que se utilizará para acotar la búsqueda. El tablero inicial le es asignado al trabajador 0, proceso que también se encarga de detectar la terminación de la búsqueda.

Un proceso que posee trabajo en su lista abierta, a lo sumo procesa una cantidad fija de nodos en cada iteración (pasada como parámetro al algoritmo) o procesa nodos hasta encontrar una solución o que se vacíe su lista abierta. A continuación el trabajador recibe, si los hay, costos de "mejores soluciones" encontradas por los demás y a medida que esto ocurre actualiza (si es necesario) su variable CMS. De esta manera los nodos a procesar serán sólo los que tengan costo menor a CMS.

Si el proceso todavía posee trabajo en su lista abierta, entonces examina si otros procesos le hicieron pedidos de trabajo, caso en el cual envía nodos del principio y final de su lista abierta al trabajador solicitante ocioso. Luego continúa trabajando sobre sus nodos.
En caso contrario, el proceso está ocioso, por lo que envía un pedido de trabajo a su donador siguiendo el algoritmo ARR.
Si el proceso encuentra una nueva solución, envía a todos los demás procesos el costo de la misma.
Luego espera los siguientes tipos de mensajes, que serán atendidos sin prioridad alguna:

- *Petición de trabajo:* un trabajador ocioso seleccionó a este proceso como su donador.
- *Trabajo:* el donador envía el trabajo requerido. Ahora el proceso está activo nuevamente.
- *Rechazo de pedido de trabajo:* el donador seleccionado no tiene trabajo. El proceso debe enviar un mensaje de pedido de trabajo al próximo donador.
- *Token:* recepción del token para detección de terminación. Si es necesario se actualiza el token y se pasa al siguiente proceso. El proceso 0, al recibir el token, chequea la terminación.
- *Nueva solución encontrada por otro:* en caso de ser necesario, se actualiza la variable CMS.

Cuando el proceso 0 detecta la terminación, envía un mensaje a los demás procesos avisando el fin del cómputo.
Se utilizó el token de terminación para trasladar los movimientos de la solución de costo mínimo hasta el proceso 0, de forma que los mensajes de comunicación de nuevas soluciones encontradas durante el algoritmo sólo posean un valor entero, el costo, evitando así overhead de comunicación.

3. TRABAJO EXPERIMENTAL REALIZADO

Para las pruebas se dispone de un clúster homogéneo compuesto por 20 procesadores Pentium 4 de 2.4GHz y 1GB de memoria RAM.

En las pruebas realizadas en este trabajo se utilizaron diferentes tamaños de tableros: 4x4 (15-Puzzle), 5x5 (24-Puzzle) y 6x6 (35-Puzzle). Para analizar el comportamiento de la aplicación al escalar respecto a la cantidad de procesadores, cada tablero se probó con subconjuntos de P máquinas pertenecientes al cluster antes mencionado, donde P = {4, 6, 8, 12, 16}.

El trabajo a realizar en cada prueba depende del tamaño del tablero y de la configuración inicial de este, es decir de la posición en que se encuentra cada pieza al iniciar la aplicación. Cabe aclarar, que muchos de estos estados iniciales no poseen solución.

Para este trabajo se ha tomado un conjunto de configuraciones iniciales que representan el trabajo promedio de aquellas configuraciones que tienen solución, el cual ha sido obtenido en estudios anteriores [23].
Las configuraciones utilizadas en cada tamaño de tablero se muestran en la figura 2.

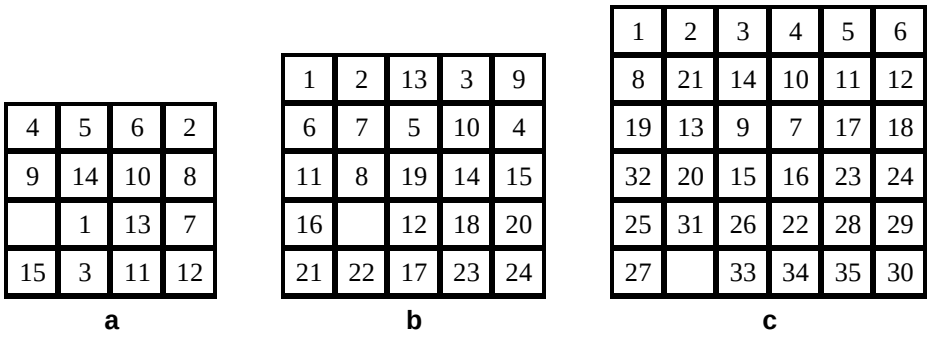


Figura 2. Configuración inicial para las pruebas con tableros (a) 4x4, (b) 5x5, (c) 6x6.

En las Tablas 1, 2 y 3 se muestran los resultados para las configuraciones de la Figura 2. En cada tabla se muestra la cantidad de procesadores usados, el tiempo medido en segundos requerido por la aplicación, el speedup alcanzado, y la pseudoeficiencia lograda.

Cantidad Máquinas	Tiempo	SpeedUp	Pseudoeficiencia
Secuencial	2769,3028	--	--
4	155,851872	17,76881298	4,442203245
6	153,272189	18,06787509	3,011312515
8	154,163291	17,9634383	2,245429787
12	140,379964	19,72719388	1,643932823
16	216,179074	12,81022587	0,800639117

Tabla 1. Resultado de las pruebas con la configuración de la Figura 2.a

Cantidad Máquinas	Tiempo	SpeedUp	Pseudoeficiencia
Secuencial	173,38081	--	--
4	11,784312	14,71284959	3,678212398
6	12,479954	13,89274432	2,315457386
8	13,015765	13,32083131	1,665103914
12	12,957742	13,38048018	1,115040015
16	10,115011	17,14094132	1,071308832

Tabla 2. Resultado de las pruebas con la configuración de la Figura 2.b

Cantidad Máquinas	Tiempo	SpeedUp	Pseudoeficiencia
Secuencial	655,099622	--	--
4	130,10891	5,035009685	1,258752421
6	41,846063	15,65498819	2,609164698
8	28,970804	22,61240737	2,826550922
12	32,510075	20,15066474	1,679222062
16	39,065716	16,76916972	1,048073108

Tabla 3. Resultado de las pruebas con la configuración de la Figura 2.c

La Figura 3 muestra el speedup alcanzado en cada prueba en un gráfico comparativo. En este se nota que en la mayoría de los casos se alcanza un speedup superlineal, el cual comienza a decrementarse al utilizar muchas máquinas debido al incremento de la interacción entre las tareas.

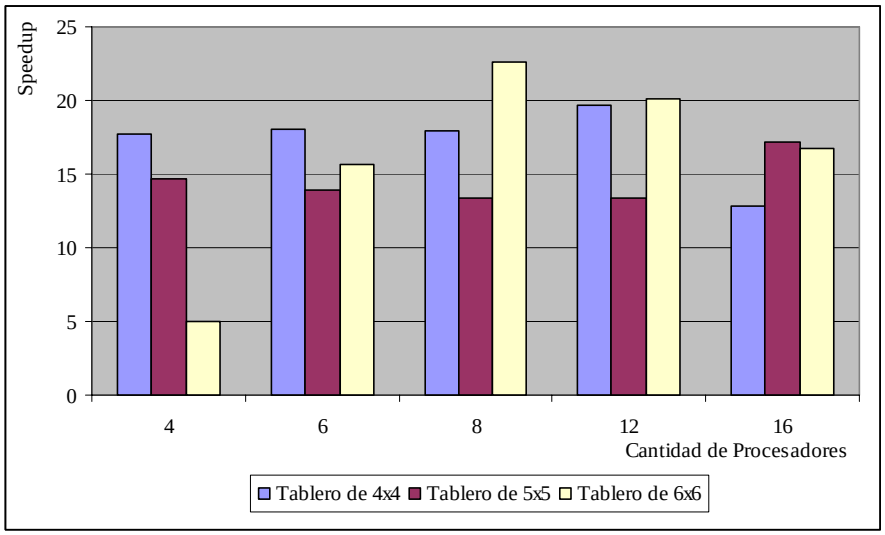


Figura 3. Gráfico comparativo del speedup de las pruebas de la tabla 1.

La Figura 4 muestra la pseudoeficiencia lograda. En este caso se puede ver como se va decrementando a medida que se agregan procesadores, llegando en algunos casos a ser menor a 1.

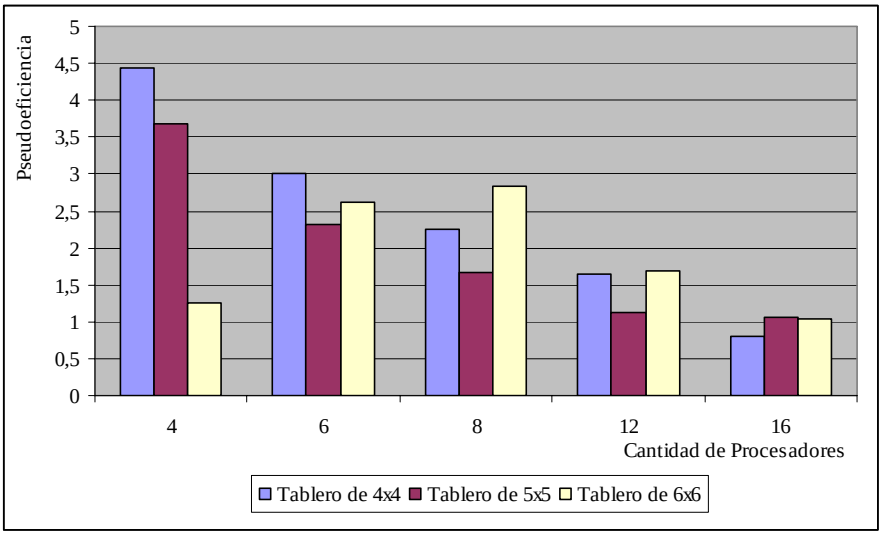


Figura 4. Gráfico comparativo de la pseudoeficiencia de las pruebas de la tabla 1.

4. CONCLUSIONES Y LINEAS DE TRABAJO FUTURO

Se ha presentado un análisis de la solución paralela para el problema del Puzzle N^2 -1, analizando Superlinealidad y Eficiencia.

Se ha estudiado la evolución de ambos parámetros en función del número de procesadores del cluster empleado.

Actualmente se están investigando métricas de estimación del trabajo a desarrollar para predecir el tiempo de respuesta en función del número de procesadores utilizados. En estos casos se han analizado previamente el espacio de estados iniciales que son posibles para alcanzar alguna solución.

Asimismo se tratará de separar los tiempos de procesamiento y comunicación, a fin de estimar la pérdida de rendimiento al pasar de una arquitectura de cluster a una de multicluster.

REFERENCIAS

[1] Grama A., Gupta A., Karypis G., Kumar V., “An Introduction to Parallel Computing. Design and Analysis of Algorithms”, Pearson Addison Wesley, 2nd Edition, 2003.

[2] Leopold C. “Parallel and distributed computing. A survey of models, paradigms, and approaches”. Wiley Series on Parallel and Distributed Computing. Albert Zomaya Series Editor, 2001.

[3] K. Hwang, "Advanced Computer Architecture. Parallelism, Scalability, Programmability", McGraw Hill, 1993.

[4] M. J. Quiin. “Parallel Computing: Theory and Practice”. McGraw-Hill Companies; 2 Sub edition (September 1, 1993).

[5] Baker M., R. Buyya. "Cluster Computing at a Glance". R. Buyya Ed., High Performance Cluster Computing: Architectures and Systems, Vol. 1, Prentice-Hall, Upper Saddle River, NJ, USA,pp.3-47,1999.

[6] Barry Wilkinson, Michael Allien. “Parallel Programming: Techniqus and Applications Using Network Workstation and Parallel Computers”. Pearson Prentice Hall, 2nd Edition, 2005.

[7] D. P. Helmbold and C. E. McDowell, “Modeling speedup (n) greater than n,” in Proc. Int. Conj Parallel Processing, 1988, pp. 8-12.

[8] Vasco M. Manquinho and João P. Marques-Silva. “Search Pruning Techniques in SAT-Branch-and-Bound Algorithms for Covering Problem”. IEEE Transactions on computer- Aided design of integrated circuits and systems. Vol. 21, Nº. 5, May 2002.

[9] I. V. Sergienko, V. P. Shylo. “Problems of discrete optimization: Challenges and main approaches to solve them”. Cybernetics and Systems Analysis. Springer New York. Volume 42, Number 4 / July, 2006. Pages 465-482.

[10] Hart Lambur, Blake Shaw. “Parallel State Space Searching Algorithms”. May 2004.

[11] Grama, A.; Kumar, V. ”State of the art in parallel search techniques for discrete optimization problems”. Knowledge and Data Engineering, IEEE Transactions. Pages: 28-35, Volume: 11, Issue: 1, Jan/Feb 1999.

[12] A. Ferreira and P. Pardalos. “Solving Combinatorial Optimization Problems in Parallel: Methods and Techniques”. Lecture Notes in Computer Science No. 1,054, State-of-the-Art Surveys, Springer-Verlag, 1996.

[13] Rao V. N. and V. Kumar. "On the efficiency of parallel backtracking". IEEE Transactions. Parallel and Distributed System, (4)4:427-437.

[14] Korf, R.. “Large-scale parallel breadth-first search”. Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-2005) (Pittsburgh, PA), 1380--1385. 2005.

[15] T. Anderson, D. Culler, D. Patterson, NOW Team, “A Case for NOW (Networks of Workstations)”, IEEE Micro, 15(1), 1995, pp. 54-64.

[16] Bohn C., Lamont G., "Load Balancing for Heterogeneous Clusters of PCs", Future Generation Computer Systems, Elsevier Science B.V., Vol 18, 2002, pp 389-400.

[17] J. Basney, M. Livny. "Deploying a High Throughput Computing Cluster". R. Buyya Ed., High Performance Cluster Computing: Architectures and Systems, Vol. 1, Prentice-Hall, Upper Saddle River, NJ, USA, pp. 116-134, 1999.

[18] D. Ratner and M. K. Warmuth. “The (n2 − 1)-puzzle and related relocation problems”. Journal for Symbolic Computation, 10:11–137, 1990.

[19] Parberry, I. A Real Time Algorithm for the (n2 – 1) Puzzle. Information Processing Letters, Vol. 56, pp. 23-28, 1997.

[20] M. Gardner. “The Mathematical Puzzles of Sam Loyd”. Dover, 1959.

[21] Plamenka Borovska. “Parallel Combinatorial Search on Computer Cluster: Sam Loyd’s Puzzle”. International Conference on Computer Systems and Technologies - CompSysTech’06

[22] E. W. Dijkstra and CS.cS. holten. “Termination detection for diffusing computations. Information Processing Letters, 11:1-4, 1980.

[23] Victoria Sanz. Informe Técnico “Paralelización de N-Puzzle”. Junio 2007.

Using Parallel Pivot vs. Clustering-based Techniques for Web Engines

Gil-Costa Veronica and Marcela Printista
DCC, Universidad de San Luis,
San Luis, Argentina
{gvcosta,mprinti}@unsl.edu.ar

July 25, 2007

Abstract

Web Engines are a useful tool for searching information in the Web. But a great part of this information is non-textual and for that case a metric space is used. A metric space is a set where a notion of distance (called a metric) between elements of the set is defined. In this paper we present an efficient parallelization of a pivot-based method devised for this purpose which is called the *Sparse Spatial Selection* (SSS) strategy and we compare it with a clustering-based method, a parallel implementation of the *Spatial Approximation Tree* (SAT). We show that SAT compares favourably against the pivot data structures SSS. The experimental results were obtained on a high-performance cluster and using several metric spaces, that shows load balance parallel strategies for the SAT. The implementations are built upon the BSP parallel computing model, which shows efficient performance for this application domain and allows a precise evaluation of algorithms.

Keywords: Metric Space, BSP, Parallel Search, Distance Computations.

1 Introduction

The World Wide Web has information useful to millions of people. Some simple browse the Web through entry points such as Yahoo!. But many information seekers use a search engine to begin their Web activity. In this last case usually users submit a query of keywords, and receive a list of Web pages containing the keywords that may be relevant. There is no question that the Web is a huge and challenging to deal with. For helping users to find their answers, the search engine module is responsible for receiving and filling search request from users. The engine relies heavily on the indexes and some time on the page repository. Because of the Web's size, and the fact that users typically only enter one or two keywords, result sets are usually very large. Therefore the ranking operation has the task of sorting the results. The query module is of special interest because traditional information retrieval (IR) techniques have run into selectivity problems when applied without modifications to Web searching: most traditional techniques rely on measuring the similarity of query texts with texts in a collection's documents.

With the growth of non-text content of the Web, it is becoming increasingly important to store, index and search over images, audio, and video collections. Metric spaces and similarity search are used for that kind of objects. The computational cost of the algorithms that determine the similarity

between two objects makes similarity search an expensive operation and a case for its efficient parallelization. This fact has motivated the development of many research works aiming to do efficient similarity search over very large collections of data.

Many research studies have been presented so far about multimedial structures. Some of them are Burkhard-Keller-Tree (BKT) [5], Fixed-Queries Tree (FQT) [2], Fixed-Height FQT (FQHT) [1], Fixed-Queries Array (FQA) [6], Vantage Point Tree (VPT) [22], Approximating and Eliminating Search Algorithm (AESA) [21], LAESA (Linear AESA) [12], Bisector Trees (BST) [9], Generalized-Hyperplane Tree (GHT) [17], Geometric Near-neighbor Access Tree (GNAT) [3] and Spatial Approximation Tree (SAT) [13]. These structures are used to perform similarity searches in a *metric space*.

In this paper we present a parallel strategy for the Sparse Spatial Selection (SSS) index which is a pivot based-technique. We present strategies to reduce the number of distances computations and the I/O requirements. We compare the efficiency of the algorithms implemented with the parallel implementation of a clustering-based technique, the Spatial Approximation Tree (SAT). We use the Bulk Synchronous Parallel - BSP [20] model to perform the parallel querying and data distribution.

The rest of this paper is organized as follow. Section 2 introduces the theoretical concepts needed to understand the problem. In Section 3 the sequential SSS index is presented. Section 4 presents the Sequential SAT algorithm. In Section 5 the parallel platform is explained. In Section 6 we present the parallel SSS index strategies and in Section 7 the parallel SAT algorithms. Section 8 preset the results obtained and finally Section 9 presents the conclusions and future works.

2 Theoretical Concepts

A *metric space* $(\mathbb{X}, d)$ is composed of a universe of valid objects $\mathbb{X}$ and a *distance function* $d : \mathbb{X}\times\mathbb{X} \rightarrow \mathbb{X}^+$ defined among them. The distance function determines the similarity or distance between two given objects. The goal is given a set of objects and a query, retrieval all objects close enough to the query. This function holds several properties: strictly positiveness ($d(x, y) > 0$ and if $d(x, y) = 0$ then $x = y$), symmetry ($d(x, y) = d(y, x)$), and the triangle inequality ($d(x, z) = d(x, y) + d(y, z)$). The finite subset $\mathbb{U} \subset \mathbb{X}$ with size $n = |\mathbb{U}|$, is called dictionary or database and represents the collection of objects where searches are performed.

A k-dimensional vector space is a particular case of metric space in which every object is represented by a vector of k real coordinates. The definition of the distance function depends on the type of objects we are managing. In a vector space, d could be a distance function of the family L_s, defined as $L_s(x, y) = (\sum_{1\leq i\leq k} |x_i - y_i|^s)^{\frac{1}{s}}$ [4]. For example, $s = 2$ yields Euclidean distance, that is the number of insertions, detentions or modifications we have to performe to make two words equal.

The computational cost of the algorithms that determine the similarity between two objects makes similarity search an expensive operation and a case for its efficient parallelization. This fact has motivated the development of many research works aiming to do efficient similarity search over very large collections of data. Data parallel programming is particularly convenient for two reasons. The first is its easiness of programming. The second is that it can scale easily to large problem sizes.

There are two main queries of interest for a collection of objects in a metric space:

- *range search:* that retrieves all the objects $u \in \mathbb{U}$ within a radius r of the query q, that is: $\{u \in \mathbb{U}/d(q, u) = r\}$;
- *nearest neighbor search:* that retrieves the most similar object to the query q, that is $\{u \in \mathbb{U}/\forall v \in \mathbb{U}, d(q, u) = d(q, v)\}$;

In this paper we are devoted to range queries. Nearest neighbor queries can be rewritten as range queries in an optimal way [8], so we can restrict our attention to range queries. The evaluation of the distance function is very expensive, and therefore searches become inefficient if the collection has a high number of elements. Thus, reducing the number of evaluations of the distance function is the main goal of the methods for similarity search in metric spaces. To do that, they first build indexes over the whole collection. Later, using the triangle inequality, those indexes permit to discard some elements without being necessary to compare them against the query.

Search methods can be classified into two types [7]: *pivot-based* and *clustering-based* techniques. Pivot-based search techniques choose a subset of the objects in the collection that are used as pivots. The index is built by computing the distances from each pivot to each object in the database. Given a query (q, r), the distances from the query q to each pivot are computed, and then some objects of the collection can be directly discarded using the triangle inequality and the distances precomputed during the index building phase. Being $x \in \mathbb{U}$ an object in the collection, we can discard x if $|d(p_i, x) - d(p_i, q)| > r$ for any pivot p_i, since by the triangle inequality, if this condition is true, its distance to q will be $d(x, q) > r$. The objects that can not be discarded by this condition make up the candidate list, and they must be compared against the query. The total complexity of the search is the sum of the internal complexity, the comparisons of q with each pivot, and the external complexity, the comparisons of q with each object in the candidate list.

Clustering-based techniques split the metric space into a set of equivalence regions each of them represented by a cluster center. During searches, whole regions can be discarded depending on the distance from their cluster center to the query. Two good surveys can be found in [8] and [23].

A recent pivot-based technique is the *Sparse Spatial Selection* (SSS) [4]. SSS is a dynamic method since the collection can be initially empty and/or grow later. It works with continuous distance functions and it is suitable for secondary memory storage. The main contribution of SSS is the use of a new pivot selection strategy. This strategy generates a number of pivots that depends on the intrinsic dimensionality of the space. Moreover, this pivot selection strategy is dynamic since it adapts the index when new objects are added to the collection.

A recent clustering-based technique is the *Spatial Approximation Tree* (SAT) devised to support efficient searching in high dimensional metric spaces [13]. This structure has been compared successfully against other data structures [15] and update operations have been included in the original design [14]. The SAT is a nice example of tree data structure in which well-known tricks parallelization simply do not work [11, 10]. It is too sparse, unbalanced and its performance is too dependent on the workload generated by the queries being solved by means of searching the tree.

3 Sequential SSS Index

Let $(\mathbb{X}, d)$ be a metric space, $U \subset \mathbb{X}$ an object collection, and M the maximum distance between any pair of objects, $M = \max\{d(x, y)/x, y \in \mathbb{X}\}$. The set of pivots contains initially only the first object of the collection. Then, for each element $x_i \in \mathbb{U}, x_i$ is chosen as a new pivot if its distance to every pivot in the current set of pivots is equal or greater than $\alpha\, M$, being α a constant parameter. That is, an object in the collection becomes a new pivot if it is located at more than a fraction of the maximum distance with respect to all the current pivots. For example, if $\alpha = 0.5$ an object is chosen if it is located farther than a half of the maximum distance from the already selected pivots. The following pseudocode summarizes the pivot selection process:

PIVOTS $\leftarrow \{x_1\}$
for all $x_i \in \mathbb{U}$ **do**

if $\forall\, p \in$ PIVOTS, $d(x_i, p) \geq \alpha\, M$ **then**
PIVOTS = PIVOTS $\leftarrow\ \cup \{x_i\}$

A key observation here is that the calculations performed to obtain the values of the distance function $d(x_i, p)$ during the construction of the SSS index are not discarded, they actually form the index itself. Namely for each pivot, the SSS index maintains the distance between each database object and all of the pivots. Thus solving the range query (q, r) takes the following steps:

foreach pivot p **do** dq[p]$\leftarrow\ d(q, p)$
$n \leftarrow\ 0$
foreach object o **do**
foreach pivot p **do**
if (distance[o][p] $>$ (dq[p]$-r$) **and**
distance[o][p] $<$ (dq[p]$+r$)) **then**
$n \leftarrow\ n + 1$
endif
endfor

if (n = total number of pivots) **then**
add object o to a list of candidate objects ℓ.
endif
endfor
foreach object $o \in \ell$ **do**
if ($d(o, q) \leq r$) **then**
report object o as solution
endif
endfor

It seems evident that all the selected pivots will not be too close to each other. Forcing the distance between two pivots to be greater or equal than $M\alpha$, ensures that they are well distributed in the whole space. It is important to take into account that the pivots are not very far away from each others neither very far from the rest of objects in the collection (i.e., they are not *outliers*), but they are well distributed covering the whole space. The hypothesis is that, being well distributed in the space, when a search is performed the set of pivots will be able to discard more objects than pivots selected with a different strategy.

Being dynamic and adaptive is another good feature of the pivot selection technique. The set of pivots adapts itself automatically to the growing of the database. When a new element x_i is added to the database, it is compared against the pivots already selected and it becomes a new pivot if needed. In this way the number of pivots does not depend on the collection size but on its intrinsic dimensionality of the metric space. Actually the collection could be initially empty, which is interesting in practical applications.

Although in this method it is not necessary to state in advance the number of pivots to use, it is necessary to set the value of α. This value determines the number of pivots. It is clear that the bigger the value of α, the smaller the number of pivots that can be "placed" into the space. However, α must always take values between 0.35 and 0.40, depending on the intrinsic dimensionality of the space. That is, the optimal results in SSS are always obtained when α is set to those values and in general a higher α works better when the intrinsic dimensionality is higher [4].

4 Sequential SAT

The SAT construction starts by selecting at random an element a from the database $S \subset U$. This element is set to be the root of the tree. Then a suitable set $N(a)$ of neighbours of a is defined to be the children of a. The elements of $N(a)$ are the ones that are closer to a than any other neighbour. The construction of $N(a)$ begins with the initial node a and its bag holding all the rest of S. We first sort the bag by distance to a. Then we start adding nodes to $N(a)$ (which is initially empty). Each time we consider a new node b, we check whether it is closer to some element of $N(a)$ than to a itself. If that is not the case, we add b to $N(a)$. We now must decide in which neighbour's bag we put the rest of the nodes. We put each node not in $a \cup N(a)$, but in the bag of its closest element of $N(a)$. The process continues recursively with all elements in $N(a)$.

The structure is a tree that can be searched for any $q \in S$ by spatial approximation for nearest neighbour queries. The mechanism consists in comparing q against $a \cup N(a)$. If a is closest to q, then a is the answer, otherwise we continue the search by the subtree of the closest element to q in $N(a)$.

It is a little interest to search only for elements $q \in S$. The tree we have described can, however, be used as a device to solve range queries for any $q \in U$ with radius r. The key observation is that, even if $q \notin S$, the answer to the query are elements $q' \in S$. So we use the tree to pretend that we are searching an element $q' \in S$. Range queries q with radius r are processed as follows. We first determine the closest neighbour c of q among $\{a\} \cup N(a)$. We then enter into all neighbours $b \in N(a)$ such that $d(q,b) \leq d(q,c) + 2r$. This is because the virtual element q' can differ from q by at most r at any distance evaluation, so it could have been inserted inside any of those b nodes. In the process we report all the nodes q' we found close enough to q. Finally, the covering radius $R(a)$ is used to further prune the search, by not entering into subtrees such that $d(q,a) > R(a) + r$, since they cannot contain useful elements. The following pseudocode summarizes the range search:

```
SEARCH(Node a, Query q, Radius r, Dist. dmin)
if   (d(a,q) ≤ R(a)+r)   then
    if   (d(a,q) ≤ r)   then
        report a
    endif
    d_min = min{d(c,q), c ∈ N(a)} ∪ d_min
    for (b ∈ N(a)) do
        if   (d(b,q) ≤ d_min+2r) then
            RangeSearch(b,q,r,d_min)
        endif
    endfor
endif
```

5 Parallel Platform

In the BSP model of parallel computing [20], any parallel computer is seen as composed of a set of P processor local-memory components which communicate with each other through messages. The computation is organized as a sequence of supersteps. During a superstep, the processors may perform sequential computations on local data and/or send message to others processors. The messages are available for processing at their destination by the next superstep, and each superstep is ended with the barrier synchronization of processors [16].

The practical model of programming is SPMD, which is realized as C and C++ program copies running on P processors, wherein communication and synchronization among copies are performed by ways of libraries such as BSPlib [18] or BSPpub [19].

We choose the BSP model because it is deadlock-free and has a particular way of organizing computations and the resulting performance is in fact not too far from the one obtained with fully asynchronous message passing realizations. Its main advantage stems from the fact that BSP provides a cost model that allows a seemingly precise evaluation of the communication and computation costs of parallel programs. In this particular work, we have used the BSPonMPI library that allows running BSP using the MPI primitives.

The environment selected to process the queries is a cluster of computers connected by fast switching technology. We assume a server operating upon a set of P machines, each containing its own memory. Client request are sent to a broker machine, which in turn distribute those request evenly onto the P machines implementing the server. Requests are queries that must be solved with the data stored on the P machines. We assume that under a situation of heavy traffic the server start the processing of a batch of $Q = q\,P$ queries in every superstep. Basically every processor has to deal with two kinds of messages, those from newly arriving queries coming from the broker, in which case a search is started in the processor, and those from queries located in others processors that decided to continue their search locally in this processor.

We assume that the broker distribute $Q = q\,P$ queries in every superstep so that q new queries arrive at each processor in each superstep. In this case, it is not difficult to see that the cost of the broadcast operation we employ is $O(q\,P + q\,P\,G + L)$ against the $O(q\,P + q\,P^2\,G + L)$ common practice strategy reported in the literature.

6 Parallel Strategies for the SSS Index

The SSS index is a pivot-based technique where distances between the objects in the collection and the pivots are computed before the search operation starts. It can be seen as a table with so many columns as pivots and so many rows as objects in the database. This strategy can be easily parallelizable distributing the rows among the processors. But in this case the pivots must be replicated in all processors because the query objects have to be compared against them. The replication is due to the SSS index performs an intersection between the database objects and the pivots. Therefore all objects with $d(o, piv) \in \{d(q, piv) + r, d(q, piv) - r\}$ for every pivot in the set of pivots, are selected as candidates to be part of the answer. The non-pivot objects are distributed in a multiplexed way.

This method can be seen as a global strategy because all processors share the same pivots but they have only a piece of the whole database. With this approach we can use two different strategies to perform the queries searches. In the first one called *Strategy-B*, the broker machine sends the query to one processor of the system selected in a circular way as in the first strategy. Then the processor receiving the query performs a broadcast so all processors get the same query. After receiving this query, processors search in their local SSS index the most similar objects, namely the objects that satisfies the range query (q, r) and send them to the requesting processor. This processor finally waits for all the results obtained in the previous step, and sends the results to the broker machine (see Figure 1). For the case of queries requiring the nearest k objects within distance r – we denote this operation (q, r, k) – every processor sends its k nearest objects and the requesting processor determines the best k among the $k\,P$ candidates.

The second strategy applied to this global index approach, called *Strategy-C*, is to build the results for a specific query in several steps. So many steps as processors are in the system (see Figure 1). In this scheme when a new query arrives from a broker machine, the receptionist processor performs a

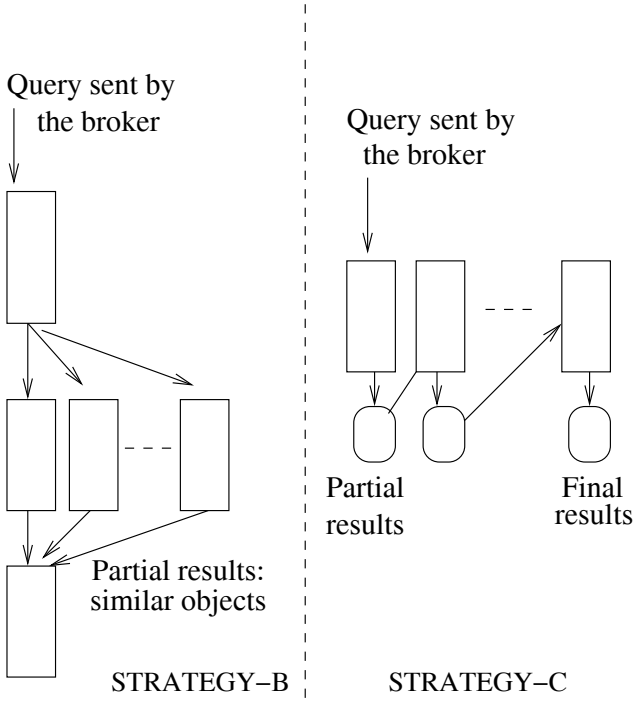


Figure 1: Global index distribution.

local similarity search using its portion of the SSS index. This processor gets a partial result for the query and sends it to the next processor. This is repeated until all processors have searched in their local indexes for similar objects. And the last one sends the final results to the broker machine. For k-nearest queries (q, r, k), together with the message travelling from one processor to the another there is an indication of the maximum distance to q of the set of K candidates contained in the message. This value is used to prone the number of distance calculations in the receiving processor.

This case is very similar to the Strategy-B, but here the communication is performed in smaller chunks. The results are passed from one processor to another until building the final results, so we have P messages with partial results for every query, and all processors are performing the same tasks in every superstep. While in the other strategy, the communication is performed in bigger chuncks and the receptionist machine has to perform the selection of the best results.

Another way to parallelize the SSS index would be to spread the space between the processors and then each one builds its own SSS index with the local data (*Strategy-A*). The problem in this strategy is that each processor will have to select new pivots locally, different from the global ones. And therefore, the number of pivots will be greater than in the other strategies proposed. To avoid the effects of imbalance from objects skewed to particular regions in space, we use multiplexed object distributed strategy, so each object is sent to the processor $p = id_{obj} mod\ P$.

To process a query using this scheme, the broker machine sends the query to one processor from the system selected in a circular way, and this processor sends a copy of the query to all other processors ($O(q\,P + q\,P\,G + L)$ broadcast cost). Then, using the local SSS indexes each processor searches for the most similar objects to the query. The results obtained are sent to the broker machine.

7 Parallel SAT Strategies

A first point to emphasize is that the SAT structure contains nodes of very diverse number of children. Every child node causes a distance comparison, so it is relevant to be able to balance the number of distance comparisons performed in every processor per superstep. In the first parallel strategy for the SAT, we propose to distribute the subtree of the root node among processors at random (R strategy). In this case the root is replicated in all processors and queries are distributed in a circular way. The processor receiving the query determines where it has to be solved.

Another parallel implementation considers the number of nodes that each processor has. So we select the processors with fewer nodes to send a subtree. Queries are distributes in a circular way and the processor receiving one query will determinate where the query must be solved. But, while we were doing some experiments to study the results obtained by this strategy (load processor strategy - LP), we realize the number of comparisons performed depends not only in the number of nodes but also in the query itself. Therefore to reduce the number of comparison distance we present another strategy to map the tree nodes onto the processors by considering the number of distance comparisons that may be potentially performed in every subtree rooted at the children of the SAT's root. That is, the subtrees associated with nodes b in $N(a)$ where a is the root and $N(a)$ is the set neighbour of a. To do that, we replicate the root and each child of the root in every processor and we distribute all the others nodes evenly through the processors, in a multiplexed way. A disadvantage is that every node has to replicate its children locally, to be able to perform the distance comparisons and in this way continue the searching operation.

In the multiplexed strategy (M), we also have to send the query to one processor and then it determinates where the query has to be solved. Therefore, we have more communication and more synchronization during a query processing operation.

These three strategies have a global distribution, because the SAT is sequentially built and then the nodes are distributed in the server. Another way to parallelize this structure (named local strategy L) is to distribute the database among the processors and then each processor builds its own local SAT structure. This case requires broadcasting the queries, because there is no communication between the processors during the query search operation and because they process these queries in a sequential way.

To improve efficiency we set an upper limit V to the number of distance comparisons that are performed per processor in each superstep. During a superstep, every time any processor detects that it has performed more than V distance comparisons, it suspends query processing and waits until the next superstep to continue with this task. Under the BSP model it means that all queries going down in a tree in each processor k has to be sent again to the processor k as a message, exactly as if it found out that the search have to continue in other processor. But no communication cost is involved for these extra messages. Also the processors stop extracting messages from its input queue. Besides, every S supersteps we collect statistics that are used to define the value of V for the next sequence of supersteps. This statistics are independent of the value of V and of the S supersteps used to calculate them. In this way the value of V can adapt itself to the workload changes produced by the flow of queries arriving constantly to the server.

Because of limit V, supersteps can be truncated before processing all the available queries. Therefore real supersteps are not a reliable measure of the real average number of supersteps required to complete a query. To deal with this, we put in every query q a counter of virtual supersteps different from the real ones executed by the BSP computer. Also, we keep a counter for the virtual supersteps in each processor k. Every time a new query is initialized in a processor k we set the virtual supersteps of the query to be equal to the number of batch it belongs to. The broker can do this before sending the

query to the processor. Besides, every time a query has to migrate to another processor we increase the virtual supersteps in one unity, because it takes one virtual superstep to get there. Additionally, we count the total number of distance calculations that has been performed in every processor k. It gives us a precise idea of global load balance (across supersteps).

8 Results

The database collection used in the experiments showed below is a 69Kwords English dictionary and a 51Kwords Spanish dictionary. This system has 32 dual processors (2.8GHz) that use NFS mounted directories. Queries were selected at random from a log of 127,000 queries. In each superstep we introduce $Q = 32$ queries per processor and each of them solves 10,000 queries. So the total number of queries processed in the system is 10,000 P.

We measure the efficiency of each strategy explained in this paper as the ratio A/B where A is the average distance computation performed in each supersptep by all processors, and B is the maximum number of distance computations performed in that superstep by any processor. This measure gives us an idea about the load balance of the system.

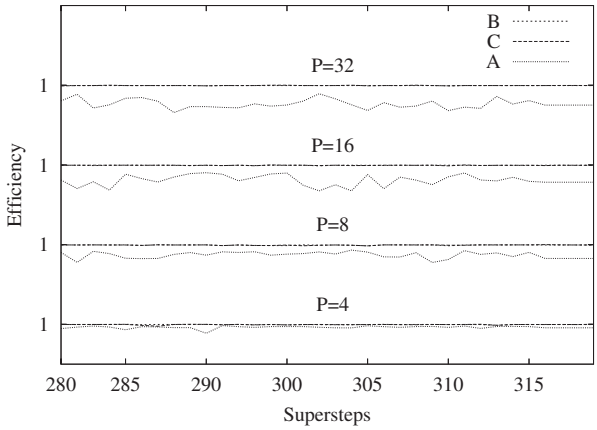


Figure 2: Efficiency obtained by the Strategy-A, Strategy-B and Strategy-C using up to 32 processors with the SSS technique.

Figure 2 shows the efficiency in each superstep. Strategies B and C obtains a good performance (they are overlapped in the graphic) and they show a better load balance than the Strategy-A. This last one presents fluctuations and the efficiency is more unstable. This is because queries requiring more distance computations may fall in the same processor, while in both others strategies the work is distributed among the processors. Figure 3 shows the efficiency obtained by the proposed parallel strategies for the SAT. In this case the L strategy is the one presenting less efficiency and therefore less load balance between supersteps.

Figure 4 shows the running time in seconds divided by the maximum running time required by the parallel implementations. The running time in this graphic increase with a greater number of processor because the amount of queries processed is also larger. With this scheme we can see if the strategies presented allows obtaining scalability. Here, we can see that Strategy-A obtains lower running times than the two others strategies based in a global distribution approach for the SSS index. This strategy only performs a broadcast at the beginning of processing each query, and then the process is performed locally and some processors concentrate a great number of distances computations

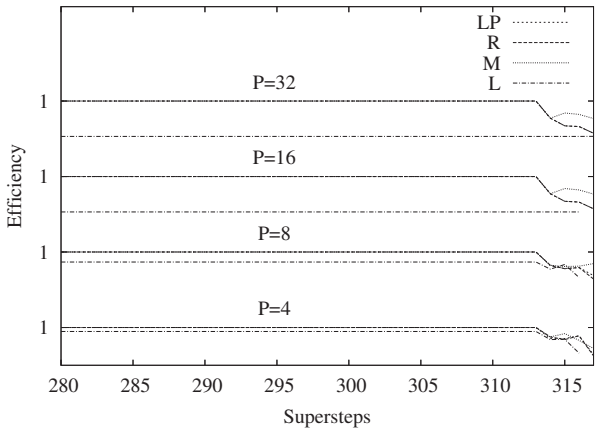


Figure 3: Efficiency obtained with 4,8,16 and 32 processors using the LP, R, M and L parallel strategies proposed for the SAT method.

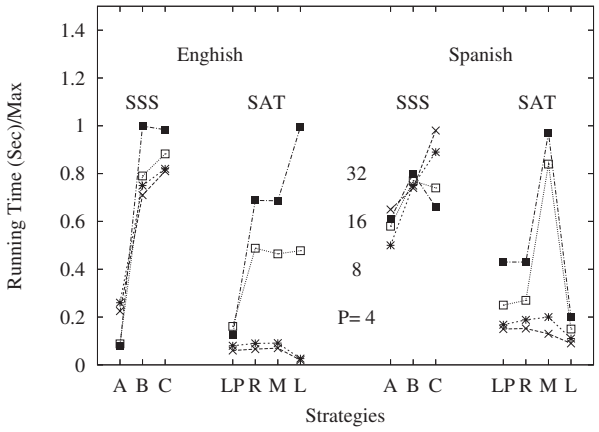


Figure 4: Running time. Each machine process $10,000P$ queries in batches of $Q = 32$. The results are presented with a system with up to 32 processors.

in some processors, while other may remind with less load work.

The parallel SAT strategies proposed used an upper limit V to balance the number of computations performed in each superstep. Figure 5 shows in the $y - axis$ the values obtained by the upper limit per superstep. This is presented for 4-32 processors using the LP, R, M and L strategies for the SAT.

Finally, Figure 6 shows in the x-axis the parallel strategies A, B and C for the SSS index and LP, R, M and L for the SAT. At the left of this graphic we have the average distance computations performed using 4, 8, 16 and 32 processors for the SSS index. In this case the B strategy is the one reporting higher values. On the other hand, the different distance computations presented by the execution of the algorithms with more processors is hard to distinguish. The average efficiency is presented. We can see that the A strategy is the one reporting less distance computations but also has less load balance due to the efficiency reported is very low.

At the right of this figure, we present the average distance computations and efficiency obtained by each strategy of the SAT method with up to 32 processors. We can see that the behaviour stands like the reported before in Figure 3, so the L strategy is the one with less uniform work distribution.

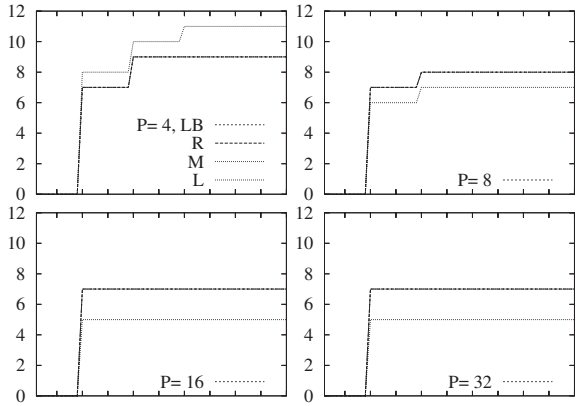


Figure 5: Upper limits adaptation for the parallel strategies of the SAT.

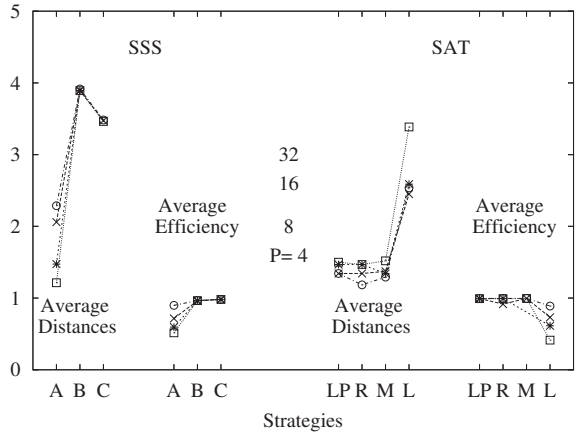


Figure 6: Average measures taken for the three strategies with 4,8,16 and 32 processors, using the English dictionary.

9 Conclusions and Future Works

In this paper we have presented the parallelization of a recent index method used to search similar objects in a metric space and we compared it with some parallel implementations of a clustering-based technique, the SAT. Results show that the SAT is more balance and obtains a better performance decreasing the number of distance computations. The SSS index allows obtaining an optimal number of pivots used to reduce the number of distance computations performed for searching similar objects.

The parallel strategies are based in two main schemes, a local scheme where the database is distributed among the processors and then each processor builds its own index or tree to perform the queries searches. The others strategies are based in a global distribution of the database. In these cases a unique SSS-index is built and then all pivots are replicated in all processors (because the intersection between queries and pivots is required by the SSS method). The database objects are distributed among the processors in a multiplexed way. In the SAT method a unique tree is built and the each node is multiplexed among processors.

As future work we intend to explore the parallel application of this method and compare it with others metric index structures like the EGNAT, and GNAT. The effect of external memory is also an interesting area applied to similarity search problems where parallel computing can allow a significant

reduction of query execution times.

References

[1] R. Baeza-Yates. Searching: an algorithmic tour. *Encyclopedia of Computer Science and Technology*, 37:331–359, 1997.

[2] R. Baeza-Yates, W. Cunto, U. Manber, and S.Wu. Proximity matching using fixed-queries trees. *In Proceedings of the 5th Annual Symposium on Combinatorial Pattern Matching*, pages 198–212, Springer-Verlag, Berlin,1994.

[3] S. Brin. Near neighbor search in large metric spaces. *In 21st conference on Very Large Databases*, 1995.

[4] N. R. Brisaboa, A. Farina, O. Pedreira, and N. Reyes. Similarity search using sparse pivots for efficient multimedia information retrieval. In *ISM '06: Proceedings of the Eighth IEEE International Symposium on Multimedia*, pages 881–888, Washington, DC, USA, 2006. IEEE Computer Society.

[5] W. A. Burkhard and R. M. Keller. Some approaches to best-match file searching. *Communications of the ACM*, 4(16):230–236, 1973.

[6] E. Chavez, J. L. Marroquyn, and G. Navarro. Overcoming the curse of dimensionality. *In European Workshop on Content-based Multimedia Indexing (CBMI99)*, pages 57–64, 1999.

[7] E. Chavez, G. Navarro, R. Baeza-Yates, and J. L. Marroquyn. Searching in metric spaces. *ACM Computing Surveys*, 3(33):273–321, 2001.

[8] E. Chavez, G. Navarro, R. Baeza-Yates, and J. L. Marroquyn. Searching in metric spaces. *ACM Computing Surveys*, 3(33):273–321, 2001.

[9] I. Kalantari and G. McDonald. A data structure and an algorithm for the nearest point problem. *IEEE Transactions on Software Engineering*, pages 631–634, 1983.

[10] M. Marin. Range queries on distributed spatial approximation trees. *International Conference on Databases and Applications (DBA'05)*, Innsbruck, Austria,Feb. 2005.

[11] M. Marin and N. Reyes. Efficient parallelization of spatial approximation trees. *International Conference on Computational Science (ICCS 2005), Lecture Notes in Computer Science 3514 (10031010), (SpringerVerlag)*, Atlanta, May 2005.

[12] L. Mico, J. Oncina, and R. E. Vidal. A new version of the nearest-neighbor approximating and eliminating search (aesa) with linear pre-processing time and memory requirements. *Pattern Recognition Letters*, 15:9–17, 1994.

[13] G. Navarro. Searching in metric spaces by spatial approximation. *n Proceedings of String Processing and Information Retrieval (SPIRE99)*, pages 141–148, IEEE CS Press, 1999.

[14] G. Navarro and N. 3. Fully dynamic spatial approximation trees. In *In Proceedings of the 9th International Symposium on String Processing and Information Retrieval (SPIRE 2002),*, 2002.

[15] S.Berchtold, C. Bohm, and D. Kein. Searching in highdimensional spaces: Index structures for improving the performance of multimedia databases. In *ACM Computing Surveys, (33(3):322373)*, 2001.

[16] D. Skillicorn, J. Hill, and W. McColl. Questions and answers about BSP. Technical Report PRG-TR-15-96, 1996. Also in *Journal of Scientific Programming*, V.6 N.3, 1997.

[17] J. K. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *AInformation Processing Letters*, 40:175–179, 1991.

[18] URL. BSP and Worldwide Standard, http://www.bsp-worldwide.org/.

[19] URL. WWW.BSP PUB Library ar Paderborn Univertity, http://http://www.uni-paderborn.de/bsp/.

[20] L. Valiant. A bridging model for parallel computation. *Comm. ACM*, 33:103–111, Aug. 1990.

[21] E. Vidal. An algorithm for finding nearest neighbors in (aproximately) constant average time. *Pattern Recognition Letters*, 4:145–157, 1986.

[22] P. Yianilos. Data structures and algorithms for nearestneighbor search in general metric spaces. *ACM Press*, pages 311–321, 1993.

[23] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity search. The metric space approach, volume 32 of Advances in Database Systems*, Springer, 2006.

Modelo y Simulación de Agentes en el Contexto de Vida Artificial aplicado al Control Aéreo en un Ambiente de Vuelo Libre

J. Ierache [1,2],Victor Battista [1], D. Rodríguez[3], P. Britos[3,2], R. García-Martínez [3,2]
1. Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica. Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales. Universidad de Morón
2. Laboratorio de Sistemas Inteligentes. Facultad de Ingeniería. Universidad de Buenos Aires
3. Centro de Ingeniería del Software e Ingeniería del Conocimiento. Escuela de Postgrado. Instituto Tecnológico de Buenos Aires
jierache@yahoo.com.ar, jierache@unimoron.edu.ar, victor2206@gmail.com, rgm@itba.edu.ar
Tel: +5411 56272000 (189/746)

Abstract

In this work one appears the results of a simulation in an artificial life environment and an approach of a model multiagents under an architecture based on beliefs, desires and intentions of the agents who implement the concept of free flight "Free Flight", this concept proposes a solution before the vertiginous growth of the aerial traffic, that today is a problematic one at world-wide level. The airways with the passage of time are multiplied to the rate of the rising world-wide economic markets. In Free Flight, the airplanes, are their own controllers, can choose their own routes of navigation, speed, height, regimes of ascent and reduction, maneuvers. For the support, analysis and management of one ambient of free flight, the characterization of a model of simulation is considered based on surroundings of artificial life in order to observe the evolution of the airspace under this concept, being considered a initials of airships with diverse intentions with respect to its destination airports.

Key words: Simulation, Artificial Life, Agents, Air Traffic Controller, Free Flight

Resumen

En este trabajo se presenta los resultados de una simulación en un ambiente de vida artificial y una aproximación de un modelo multiagentes bajo una arquitectura basada en creencias, deseos e intenciones de los agentes que implementan el concepto de vuelo libre "Free Flight", este concepto propone una solución ante el crecimiento vertiginoso del tráfico aéreo, que hoy es una problemática a nivel mundial. Las rutas aéreas con el paso del tiempo se multiplican al ritmo de los nacientes mercados económicos mundiales. En Free Flight, los aviones, son sus propios controladores, pueden elegir sus propias rutas de navegación, velocidad, altura, regimenes de ascenso y descenso, maniobras. Para el apoyo, análisis y gestión de una ambiente de vuelo libre, se considera la caracterización de un modelo de simulación basado en un entorno de vida artificial a fin de observar la evolución del espacio aéreo bajo este concepto, considerando una población inicial de aeronaves con intenciones diversas respecto a sus aeropuertos de destino.

Palabras Claves: Simulación, Vida Artificial, Agentes, Trafico Aéreo, Vuelo Libre.

1. INTRODUCCIÓN

En el campo demográfico, 14 centros urbanos del mundo tendrán más de 15 millones de habitantes; cuatro de ellos, más de 20 millones (Tokio, con alrededor de 30 millones, seguido por Lagos, Bombay y Sao Paulo) [1]. En el pasado, como en el presente, Europa, Asia Oriental y Estados Unidos son referentes económicos y generadores de tráfico aéreo. Sin embargo, en los últimos años han ganado mucho terreno la India, China y el sur de Sudamérica. Los dos primeros apoyados por una fuerte expansión económica y la última por el potencial turístico que ostenta. La aviación comercial desde sus comienzos al día de hoy ha avanzado notablemente logrando un medio de transporte seguro y efectivo. Hacia finales de febrero de 2006, la empresa Boeing anunció que había entregado 5.009 aviones de la familia '737' [2]. En la actualidad, el crecimiento del tráfico aéreo es analizado intensamente. El congestionamiento ha alcanzado límites tales que alrededor del mundo distintos pronósticos señalan el colapso de los sistemas de control de vuelo en los años próximos. Las rutas aéreas con el paso del tiempo se multiplican, el tránsito aéreo es cada vez más intenso y las pérdidas económicas a causa de los retrasos y congestionamiento son millonarias [3]. La solución clásica a este tipo de problemáticas era el incremento del número de aeropuertos y la creación de nuevas pistas de aterrizajes. Con esta perspectiva de la realidad, un cambio de enfoque en el control de tráfico aéreo surge como la solución más ventajosa dada las desventajas o implicancias de las propuestas clásicas. En el presente trabajo se describe aproximación a un modelo al control del tráfico aéreo diferente al enfoque actual en uso. La propuesta es un modelo multiagente considerando los roles mas significativos e incorporando el concepto de vuelo libre [4], [5] para la elección de las rutas de vuelo.

2. CONTROL AÉREO INTELIGENTE

El modelo de sistema multiagente que se desea desarrollar se desenvuelve en un ambiente altamente ágil y dinámico. Si bien, los componentes identificados son unos pocos, el modelo puede ser especialmente complejo de acuerdo a la cantidad de instancia de un elemento esencial: las aeronaves. Cada una de ellas, en el sistema, se modelizan a través de un agente "Piloto Aeronave". El mismo, por definición deberá mostrar un comportamiento flexible. Dicha flexibilidad le permitirá, a pesar de las distintas evoluciones que presente el ambiente, alcanzar su objetivo: lograr el aterrizaje a tiempo en su destino. Si pensamos en los distintos desafíos que un agente "Piloto Aeronave" puede encontrar durante un vuelo veremos que entran a jugar un amplio rango de eventos que pueden influenciar de forma positiva o negativa. Por ejemplo, supongamos que una aeronave parte de un aeropuerto A y se dirige a un aeropuerto B. Al principio, el agente determinará una ruta de vuelo posible según un estudio primario de sus creencias en relación al ambiente: condiciones meteorológicas, congestionamiento del aeropuerto, distancias de vuelo, rutas de vuelo tradicionales, su capacidad de combustible, entre otras. Así el agente aeronave parte con una intención/plan de vuelo para cumplir con su deseo de arribar al aeropuerto de destino en el menor tiempo con el menor consumo de combustible. Sin embargo, el sistema puede variar drásticamente, en ocasiones espontáneamente apelando a las características reactivas del Agente; otras veces, paulatinamente dando lugar a un comportamiento más pro-activo. La formación de tormentas eléctricas, la aparición de aeronaves, las condiciones meteorológicas en general, son sólo algunas de las distintas manifestaciones del ambiente en el cual se moviliza el agente "Piloto Aeronave". Finalmente, la aeronave se aproxima al espacio aéreo correspondiente al área terminal. Es en este momento, donde quizás la capacidad de negociación del agente "Piloto Aeronave" cobra vital importancia. La aeronave que se traslada de un aeropuerto A a otro B debe aterrizar. El primer problema es que la pista de aterrizaje es de por si un recurso finito y su disponibilidad sólo ocurre en pequeñas "ventanas" de tiempo. La pista de aterrizaje es un recurso que todas las aeronaves

deberán compartir en un instante de tiempo particular, sólo una de ellas podrá hacer uso de la misma para despegar o aterrizar. Esto genera una situación de competencia entre los distintos agente "Piloto Aeronave". Puede resultar que la aeronave a su llegada al aeropuerto se encuentre sólo esperando autorización para aterrizar, de modo que no tendrá problemas en hacerlo. Sin embargo, sabemos que esto no será así en las mayorías de los casos; precisamente es esta la motivación que da lugar a este trabajo de investigación. Frecuentemente, el Agente aeronave deberá llegar a un acuerdo tripartito con otros agentes aeronaves y controladores aéreos, teniendo en cuenta la disponibilidad de la pista y rutas de vuelo, el combustible con el que cuenta cada aeronave, su autonomía, afluencia de trafico, entre otros aspectos para conseguir un lugar en la cola de aterrizajes. El objetivo del sistema se centra en alcanzar la maximización del aprovechamiento de los recursos aeroportuarios en orden de aliviar la congestión en los mismos. Además, el sistema deberá gestionar los recursos para que cada uno de los aviones pueda realizar sus vuelos de forma segura. Ante el crecimiento de la congestión del tráfico aéreo la autoridad de aviación civil australiana[1] ha dado origen al programa de mejoramiento del manejo de tráfico aéreo. En un desarrollo conjunto, con el Instituto de Inteligencia Artificial Australiano, dio lugar a OASIS[2][6]. Un sistema de administración de tráfico aéreo prototipo desarrollado para el aeropuerto Kingsford Smith de Sydney. Dicho proyecto fue tomado como referencia para este trabajo. Considerando el nuevo enfoque innovador sobre la problemática, se ha considerado el concepto importante y de especial auge dentro del área del control de vuelo: el vuelo libre (Free Flight). El control de vuelo clásico, utilizado, se ha dejado de lado en favor de la implementación del vuelo libre. Los aviones, así, son sus propios controladores. Pueden elegir sus propias rutas de navegación, velocidad, altura, regimenes de ascenso y descenso, maniobras, en este contexto el objetivo del sistema es la planificación de los aterrizajes de las aeronaves como se detalla en la figura 1.El sistema le brinda total libertad al piloto para que vuele según sus convicciones, sin embargo, un sistema de control de vuelo descentralizado analiza el ambiente con el objetivo de velar por el cumplimiento del objetivo final: un vuelo seguro y sin inconvenientes. En consecuencia se pretende alcanzar un alto grado de cooperación y la capacidad de trabajar por un mismo objetivo.

Figura 1. Objetivos del sistema

3. FREE FLIGHT

Free Flight [7], propone un sistema de control que abandona el monitoreo centralizado, desde tierra, del tráfico para adoptar una arquitectura descentralizada y altamente cooperativa. El concepto descansa sobre dispositivos digitales altamente precisos para la ubicación de aeronaves (GPS: Global Positioning System), un sistema de comunicación de datos digital, que brinda información a todos los involucrados, apoyados con la automatización de gran parte de las tareas del los controladores aéreos.

[1] Denominada *Airservices Australia*

Bajo el “vuelo libre” los controladores no acaparan responsabilidades sino que ellas son delegadas a los pilotos. Los pilotos son sus propios controladores de vuelo, son libres de escoger trayectorias de vuelos, altitudes, velocidades como alguna vez lo hicieron antes de que existieran los primeros sistemas de control. El “vuelo libre” podría reducir la congestión de los aeropuertos más concurridos elevando el promedio de aterrizajes por unidad de tiempo. El concepto de free Flight [8] se define como una capacidad de operación de vuelo bajo condiciones de vuelo por instrumentos en el cual los operadores tienen la libertad de elegir su trayectoria y velocidad en tiempo real. Las restricciones al tráfico aéreo se imponen sólo para asegurar la separación, para evitar el exceso de capacidad de aeropuerto, para prevenir vuelo no autorizado a través espacio aéreo de uso restringido/prohibido, y para garantizar la seguridad del vuelo.

4. MODELO MULTIAGENTE PARA CONTROL AÉREO

El modelo propuesto se compone principalmente por cinco roles: Vigilancia y Control, Planificador, Meteorólogo, Controlador, Piloto Aeronave; según se detalla en la figura 2.

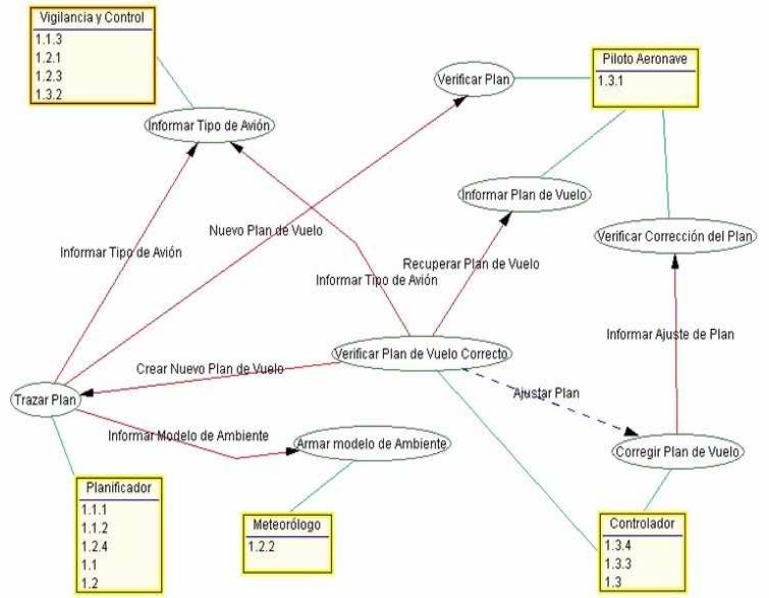


Figura 2. Modelo de roles del sistema

El rol “Vigilancia y Control” es el encargado de individualizar cada aeronave que ingresa en el espacio aéreo de su área de control. Deberá, entre otras cosas, informar la posición actual de la aeronave e identificarla.

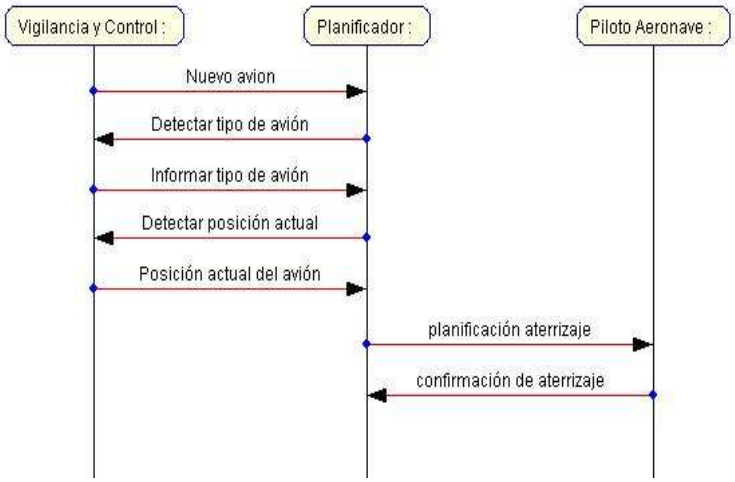
El rol “Planificador” sugiere un plan de vuelo eficaz con base en el análisis de la aeronave, las condiciones climáticas y la programación de aterrizaje. Se encarga de armar un plan de vuelo eficaz para la aeronave y lo informa como directivas similares al formato de una ‘lista de tareas’ pendientes.

El rol “Meteorólogo” actualiza la información ambiental a través de los datos que puedan recoger los distintos aviones en el aire junto con la que él mismo posee y además brindan los sensores meteorológicos del área.

El rol “Controlador” es el designado para hacer el seguimiento del desempeño de los aviones próximos al aeropuerto. Además, es el encargado de detectar cualquier tipo de inconveniente, accidente o problema que pudiera surgir, deberá corregir los planes de vuelo evaluando la situación actual de la

aeronave y, a partir de allí, componer una serie de modificaciones al plan de vuelo original. Periódicamente verificará que la trayectoria del aeronave se encuentre dentro del margen de error aceptable para el plan de vuelo estimado.
El rol "Piloto Aeronave" se encargará de tomar las decisiones de vuelo, por ejemplo, sobre: la velocidad, la altitud, la dirección. Verificará que el plan de vuelo estimado esté dentro de las posibilidades técnicas de la aeronave. Mantendrá actualizada, y disponible para el sistema, toda la información del plan de vuelo de la aeronave. Todas las correcciones al plan de vuelo original serán verificadas con el objetivo de constatar que sean compatibles con las capacidades de la aeronave.
El comportamiento de un agente aislado queda determinado por sus motivaciones individuales, sus creencias acerca del mundo y sus propias habilidades. De este modo, esta caracterización resulta insuficiente para modelar a un agente con una actitud cooperativa. La coordinación de los agentes, al igual que en otros dominios, es un aspecto importante para lograr un comportamiento coherente. La distribución de los datos y el control entre varios agentes autónomos a menudo deriva en situaciones de conflictos como consecuencia de la existencia de distintos puntos de interés, los recursos escasos y las motivaciones particulares u objetivos. El aspecto de la comunicación entre los agentes es un factor fundamental a considerar. La tendencia actual, es utilizar lenguajes de comunicación de agentes basados en la teoría de "speech acts" como por ejemplo KQML [9] y FIPA ACL [10]. En el caso de JADE es un middleware que proporciona un entorno para desarrollar agentes, que explota el potencial de la plataforma java. Otro aspecto importante es que el Framework, ofrece como servicios propios funcionalidades básicas que requiere un agente y permite centrarse en el desarrollo de los aspectos lógico de los agentes, algunas de las características que presenta JADE en cuanto a su Portabilidad: La API que proporciona JADE es independientemente de la red sobre la que va a operar, así como de la versión de Java utilizada, teniendo la misma API para J2EE [11] , J2SE [4] y J2ME [12] , Finalmente, JADE simplifica la comunicación y la cooperación entre los agentes; los agentes JADE pueden controlar su propio ciclo de vida, y pueden ser programados para que dejen de funcionar o empiecen a hacerlo dependiendo del estado del sistema y de la función que debe realizar el agente; JADE cumple con la especificación de FIPA [13], por lo tanto, puede comunicarse con agentes realizados en otros entornos pero que utilicen este estándar el Framework de desarrollo de agentes provee un amplio soporte para las comunicaciones entre agentes a partir del estándar FIPA ACL. Para ejemplificar este esquema, consideremos el caso en que dos aviones se aproximen en una área de control a un aeropuerto y se comunican con el controlador del área para establecer su lugar en la cola de aterrizajes. En este caso, ambos agentes "Piloto Aeronave" ingresan en una situación de competencia en relación a un recurso escaso, como lo es la pista de aterrizaje. Si bien éste no es un recurso agotable tiene la particularidad que puede estar disponible sólo para un agente en un instante de tiempo puntual. Cada uno de los agentes "Piloto Aeronave" tiene como uno de sus objetivos principales el ahorro de combustible y la construcción de un plan de vuelo que apunte a la maximización de este recurso. Por esta condición, al momento de ingresar al área de control, el agente "Piloto Aeronave" deberá interactuar con el agente "Vigilancia y Control" y para conseguir un lugar en la cola de aterrizajes deberá negociar con el "Planificador" quien tiene como objetivo lograr la mejor configuración del espacio aéreo para todos los agentes involucrados, según se detalla en la figura 3.

Figura 3. Diagrama de secuencias correspondiente al ingreso de un aeronave



Cada uno de los agentes que componen el sistema ha sido modelado desde la perspectiva BDI [14].El agente "Vigilancia y Control" tiene el deseo, objetivo de lograr individualizar e identificar cada aeronave que se desplaza en el espacio aéreo. Sus creencias se conforman de la información que le suministran distintos sensores de vigilancia y control del espacio aéreo. Finalmente, sus intenciones o acciones se focalizan en la transmisión de información actualizada acerca de los aviones en su espacio aéreo. El agente "Planificador" posee el deseo de programar los aterrizajes para lograr que el agente "Piloto Aeronave" pueda trazar un plan de vuelo eficaz. Sus creencias se basan en la meteorología, sus conocimientos acerca de la aeronave en cuestión, su autonomía de vuelo y la disponibilidad del recurso de la pista de aterrizaje, entre otras. Sus intenciones son la gestión de la cola de aterrizajes en el aeropuerto y sugerir al piloto las maniobras más convenientes para su aproximación dentro del área terminal con destino a su aeropuerto de aterrizaje. El agente "Meteorólogo" tiene el objetivo de actualizar la información ambiental a través de la información que puedan recoger los distintos aviones en el aire (sus creencias), apoyado con los sistemas meteorológicos, su intención es brindar un modelo de ambiente meteorológico. El agente "Controlador" debe detectar los inconveniente, potenciales incidente o problema que pudiera surgir con las rutas de vuelo de los agentes "Pilotos aeronaves". Recabará información a través de mensajes con los distintos agentes "Pilotos aeronaves" en vuelo y con el agente "Vigilancia y Control" quien le brindara la situación del espacio aéreo. Entre sus actividades se enumeran la evaluación de las rutas de vuelo y la configuración del espacio aéreo y la alerta de colisiones al agente "Piloto aeronave". El agente "Piloto Aeronave" tiene el deseo de alcanzar un aeropuerto de destino llevando adelante un vuelo seguro y eficaz sin complicaciones. Se nutre de la información que obtiene del amplio rango de instrumental de vuelo (TCAS [15], GPS) para generar sus creencias. Para alcanzar su objetivo llevará adelante verificaciones periódicas del plan de vuelo y la corrección del mismo ante una alerta de colisión.

5. MODELO DE SIMULACIÓN PARA EL CONTROL AÉREO EN AMBIENTE DE VUELO LIBRE

En este contexto según se representa sintéticamente en la figura 4, se propone una aproximación para el ambiente de vuelo libre caracterizado en un entorno de vida artificial [16], seleccionado este por su capacidad para la simulación de sistemas descentralizados donde los autómatas celulares resultan un marco para explorar fenómenos de autoorganización, presentes en el concepto de vuelo libre, donde no se cuenta con diseñador centralizado para el control aéreo. Observando las características de un hábitat

de vida artificial se considera como Depredadores a los agentes “Piloto aeronave” y como Presas se presentan dos tipos constituidos por la presa principal denominada como presa tipo pista de aterrizaje, correspondiente a su destino y la presa tipo espacio aéreo (X,Y,Z) a ocupar por un agente “Piloto aeronave” durante un tiempo (T), suficiente para asegurar las separaciones entre aeronaves y evitar colisiones, si bien son un recurso renovable como alimentos para sus depredadores, se encuentran agotados durante un margen de tiempo al ser consumidos por otro depredador representado por el agente “Piloto aeronave”. Como Hábitat de los Depredadores y Presas se considera el espacio aéreo y aeroportuario a través de parcelas o celdas organizadas en una grilla de modo similar al de un autómata celular con depredadores que deambulan por encima (aeronaves que vuelan en el espacio aéreo), cada parcela se caracteriza por la acción de los agentes Planificador, Controlador, Meteorólogo, Vigilancia y Control y la presencia de una Presa representada por el espacio aéreo que se corresponden con un juego de valores limites XYZ de la celda salvo los casos de espacios prohibidos de vuelo o restringidos temporalmente para el uso en los cuales no se consideran presas, las celdas podrán estar ocupadas por depredadores agentes “Piloto aeronave” respetando los liftes de separación entre aeronaves considerados para garantizar la seguridad aérea, y también por presas del tipo pista de aterrizaje. La energía de los depredadores (agentes “Pilotos aeronaves”) se considera en función del consumo de combustible, pudiendo agotarla o no para poder cumplir su vuelo en los márgenes de tiempo establecidos, ocurriendo la muerte del depredador si se agota su combustible o aterriza fuera de los márgenes de tiempo establecidos. El observador mira a lo depredadores (agentes “Pilotos aeronaves”) y administrar las tazas de creación de nuevos depredadores y su agotamiento en función de los márgenes de tiempo que establece para el consumo de su presa, (representado por el arribo de la aeronave a su pista del aeropuerto de destino), además el observador monitorea la actividad de las presas y de las parcelas o celdas del hábitat (espacio aéreo). La extinción de la especie se daría por el agotamiento de la energía de los depredadores, y el consumo de sus presas en un instante de tiempo de acuerdo a los márgenes seleccionados por el observador para lo cual los depredadores agentes “Pilotos aeronaves” no puedan consumir a su presa elegida (espacio aéreo X,Y,Z durante un tiempo T, pista de aterrizaje del aeropuerto de destino).

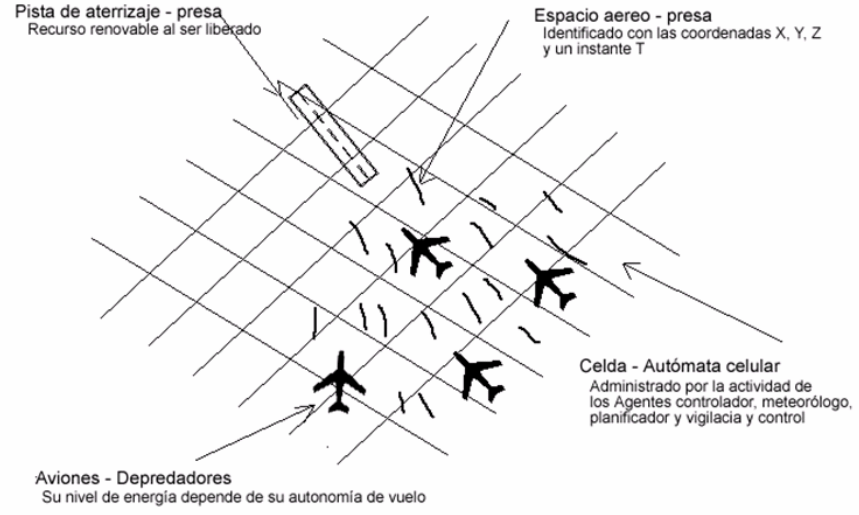


Figura 4. Ambiente de vuelo libre modelado en un entorno de vida artificial.

Para el desarrollo del modelo se analizaron herramientas orientadas a agentes como JADE [17], la especificación FIPA ACL [10] para mensajes interagentes, Protegè [18] para crear una ontología del sistema, MaSE [119] como metodología., StarLogo [20] para vida artificial. En Jared et al [21] presenta los resultados de una simulación basada en teoría de juegos para la toma de decisiones en vuelo libre por parte de las aeronaves considerando una actitud altruista especifica para la negociación en un ambiente multiagente a fin de evitar incidentes aéreos entre aeronaves en un entorno de vuelo libre, sin embargo en nuestra propuesta se considera que las aeronaves cuentan con sistemas TCAS (sistema de alerta de colisión entre aeronaves) para alertar incidentes aéreos en su espacio aéreo y realizar las maniobras de evasión a fin de mantener una separación vertical y horizontal considerada para las aeronaves que conforman el hábitat de simulación en un entorno de vida artificial. La Simulación implementada en nuestra propuesta se recrea en un ambiente de vida artificial, que se caracteriza por:

- Permitir la simulación en nuestro caso de diez aviones en competencia por los aeropuertos, la simulación se puede llevar adelante con dos (2) o tres (3) aeropuertos.
- Energía Máxima de los aviones. Este concepto es análogo a la cantidad de combustible que el avión posee para realizar su vuelo. Máximo parametrizable: cuatrocientos (400)
- Consumo de energía. Es la cantidad de energía que un avión consume al viajar una (1) unidad de distancia en busca de alcanzar su destino.
- Capacidad del aeropuerto. Es la cantidad de aviones que podrán permanecer en un aeropuerto, en un instante de tiempo dado, antes de volver a despegar. Máximo diez (10).

El Hábitat Simulado evoluciono enmarcado en las siguientes reglas:

- Los aviones tienen como objetivo lograr su aterrizaje en un destino conocido.
- Los aviones tienen un consumo de energía ligado a la distancia recorrida en busca de un aeropuerto. Al acabarse dicha energía si el avión no alcanza su destino principal o alternativa, sufre el siniestro.
- Los aviones no vuelan a una altura predefinida. Cada avión elige su altitud arbitrariamente.
- Cuando un avión llega a destino sólo hará el aterrizaje si la capacidad del aeropuerto no está colmada. De ser así, el avión recibe un nuevo destino de alternativa.
- Cuando el avión logra el aterrizaje permanecerá en “reposo” en el aeropuerto durante un tiempo, equivalente al que estuvo en vuelo, con el objetivo de recuperar la energía perdida. Mientras el avión permanece en el aeropuerto ocupa una plaza de capacidad disponible del aeropuerto.
- Los aviones en maniobras de despegue o de aterrizaje demoran un determinado tiempo “X” ocupando una plaza de capacidad disponible del aeropuerto, tiempo en el cual el recurso no puede ser explotado por otro avión.
- Los aviones en vuelo deben respetar un límite de separación, tanto vertical como horizontal. Esto define un área de seguridad que permite que el avión vuele sin riesgos de colisión.

Se presenta en la figura 5 la pantalla principal del simulador, en la que se genera el ambiente para iniciar la simulación utilizando los valores parametrizables detallados.

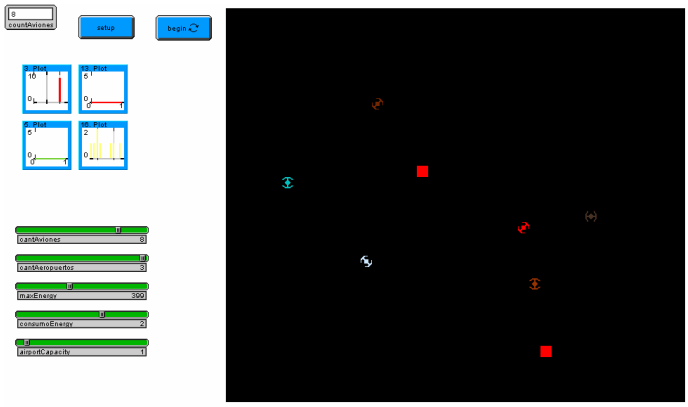


Figura 5. Pantalla de simulación en Starlogo.

La interfase de representación de la simulación cuenta, además con un monitor ubicado en el sector superior para medir el comportamiento de las aeronaves en su hábitat en tiempo real, brindando información en primer lugar: relacionada con la disponibilidad de combustible/energía de los aviones, según se detalla en la figura 6. El segundo se brinda información de la evolución de la separación vertical promedio calculada a partir de la separación con respecto al avión más próximo que registra cada avión, según se detalla en la figura 7

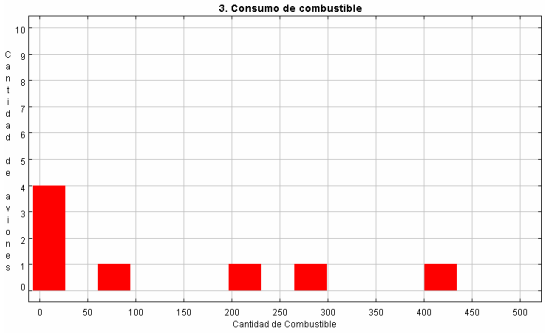


Figura 6. Disponibilidad de combustible.

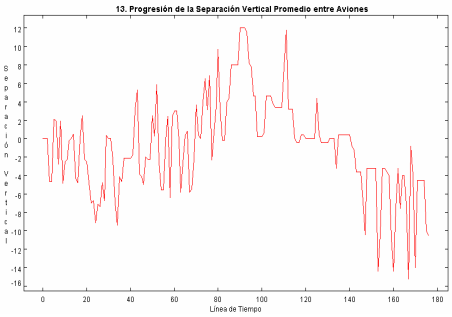


Figura 7. Evolución de la separación vertical

El tercer lugar se brinda información de la evolución de la separación horizontal promedio calculada a partir de la separación con respecto al avión más próximo que registra cada avión. Según se detalla en la figura 8.

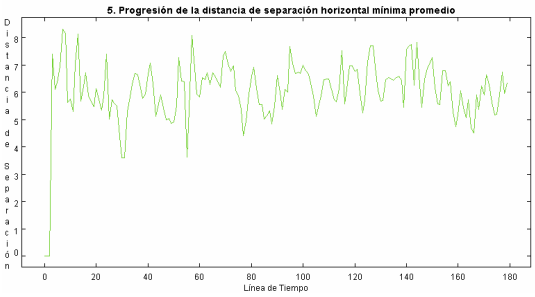


Figura 8. Evolución de la separación horizontal

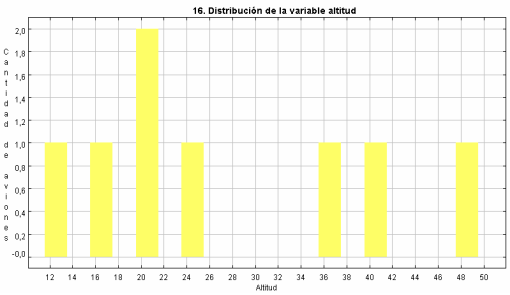


Figura 9. Distribución de la altitud entre aviones.

El cuarto lugar se brinda información de la distribución de la variable altitud de los diferentes aviones en vuelo. Según se detalla en la figura 9

7. RESULTADOS

Nuestro simulador han sido puesto a prueba en ambientes de simulación similares a los que otros investigadores han utilizado [22], [23], [24], [25]. El simulador elegido es el Starlogo dado que nos permite trabajar con ambientes densamente poblados sin problemas. Si bien es cierto que el potencial de esta herramienta es muy alto, también es cierto que dada la alta demanda de recursos de hardware necesarios para la simulación, algunas de las restricciones que hemos impuesto para nuestro modelo se deben a esta causa. Entre ellas, podemos citar la cantidad máxima de aviones o aeropuertos a simular.
El espacio aéreo, una de las "presa", es simulado a través de las "celdas" definidas en la grilla de simulación. A su vez, la "presa" aeropuerto, es también identificada por una "celda" pero particularmente diferente a las anteriores. Luego, los aviones serán agentes que reemplazarán las conocidas tortugas del simulador. Las distancias recorridas por los aviones se miden en cantidades de celdas, y el tiempo se mide en ciclos de reloj. Por cada ciclo de reloj el ambiente artificial evoluciona bajo las reglas que lo enmarcan. A pesar de que el ambiente de simulación está preparado para soportar aviones volando a distintas alturas, para llevar adelante las pruebas se restringirá la altura de vuelo a un valor único, igual para todas las instancias de agentes aviones. Así, los resultados podrán ser comparables a los obtenidos en otras investigaciones ya mencionadas. Además, como otros autores señalan, de esta forma, la facilidad para generar ambientes densamente poblados es mucho mayor, y nos brinda la posibilidad de experimentar la robustez y confiabilidad del modelo planteado. Los aviones, por cada ciclo de reloj, se trasladan hacia alguna celda contigua a la que ocupan. Cada una de las decisiones que el avión deba tomar acerca de su orientación y dirección de vuelo, se hace efectiva de forma inmediata y en el instante anterior a su traslación a su próxima celda. Los aviones cuentan con una "zona de exclusión" a su alrededor. Dicha zona, idealmente, libre de aviones, marca un sector de potencial peligro de colisión para las aeronaves. Los aviones escogen libremente su ruta de vuelo, sin embargo, cuando su ruta los lleva a ingresar en la "zona de exclusión" de otro avión, una señal de colisión se dispara y comienzan las acciones de evasión.

7.1 Escenario de espacio aéreo abierto

Este escenario se basa en la experiencia [22], y refleja el vuelo a cielo abierto. Los aviones no tienen otros obstáculos más que otros aviones y todos tienen un destino que alcanzar. En la experiencia original, como unidad de medida había sido elegida la milla. En nuestra experiencia, para poder generar una escala de equivalencias, cada milla será simbolizada por una "celda". La disposición inicial de los aviones genera un círculo en cuyo interior se apostarán los distintos puntos de destinos de los aviones. Cada avión se instancia con una zona de seguridad a su alrededor de modo que pueda ingresar al ambiente de simulación libre de potencial peligro de colisión. La tabla 1 muestra los resultados promedio en un esquema de pruebas similar al estudiado por Hill. Aún en las densidades de tráfico más, ningún incidente fue reportado. Los incidentes, y alarmas de colisiones en los escenarios de baja densidad eran esporádicos y sólo se hizo presente una fuerte tendencia en alza a partir de que las poblaciones alcanzaban los 70 aviones.

7.2 Punto de intersección

Este escenario, descrito en [23], propone un punto de intersección entre dos rutas de vuelo definidas para un conjunto de aviones. Cada flujo de aviones, sigue una ruta a lo largo de uno de los ejes cartesianos de la grilla de simulación; ya sea el eje "X" o el eje "Y". De esta forma, en punto fijo ambas rutas confluyen materializando una zona de riesgo de colisión para los aviones. Los mismos, inicialmente parten de un punto fijo de la grilla y se les asigna un destino al lo otro lado de la grilla, de

modo que suponga una ruta de vuelo en línea recta. En todos los casos en que fue realizada la prueba, no se han constatado colisiones entre los aviones. Por el contrario, se divisó un bajo nivel de alarma de incidentes para el rango de aviones inferior al los 20 aviones y una alta eficacia al igual similar a la que se había podido experimentar en la simulación a espacio abierto. Sin embargo, se encontró un punto de inflexión en el cual los resultados marcan una leve tendencia al crecimiento tanto de las alarmas de incidentes como de la pérdida de eficiencia. Este patrón de comportamiento se comprobó como resultado de la disminución del delta de espacio de separación de las naves, a la hora de su instanciación, para alcanzar los límites exigidos por la prueba.

Aviones	Incidentes	Eficiencia (%)
> 20	25 – 35	< 98.5
< 20	< 15	99.0 – 99.5

Tabla 1. Resultados de simulación de espacio aéreo abierto

Aviones	Incidentes	Eficiencia (%)
90	0/20	97.0
80	0/15	97.1
70	0/9	97.6
60	0/1	97.9
50	0/0	98.1

Tabla 2. Resultados de simulación de Punto de Intersección

7.3 Escenario “Same Point - Same Time”

Este no es un escenario real, pero bien sirve para estudiar la reacción del sistema en condiciones delicadas que exigen un sistema robusto. Para la prueba, la población de aviones se dispone formando un círculo y a cada avión se le asigna un punto de destino al otro lado del círculo obligándolo a trazar una ruta de vuelo ideal a través del centro del mismo. De esta forma, si todos ellos siguieran esta ruta, alcanzarían el punto central del círculo al mismo tiempo desencadenando la tragedia. La figura “10” muestra la evolución del ambiente artificial. El trabajo de [25], para esta experiencia, describe que durante la simulación se forman patrones de organización de los aviones en forma de “olas”. En nuestro caso, dichos patrones no se generan, sin embargo, se consigue un patrón distinto. En principio, los aviones se acercan hacia el interior del círculo como, sin advertir el conflicto. Luego, cuando el conflicto es tangible, los aviones se empiezan a entrelazar desdibujando toda disposición geométrica. Una vez que los aviones logran atravesar el centro del círculo, se vuelven a trasladar en formación geométrica hacia su destino.

Aviones	Incidentes	Eficiencia (%)
32	0/29	90.4
28	0/23	91.7
24	0/17	93.4
20	0/13	95.0
16	0/10	97.0
12	0/7	97.5

Tabla 3. Resultados de simulación de Same Time"

Vuelos	Incidentes	Eficiencia (%)
800	8	97.0
1300	0/10	97.4

Tabla 4. Resultados de Simulación de Aeropuertos

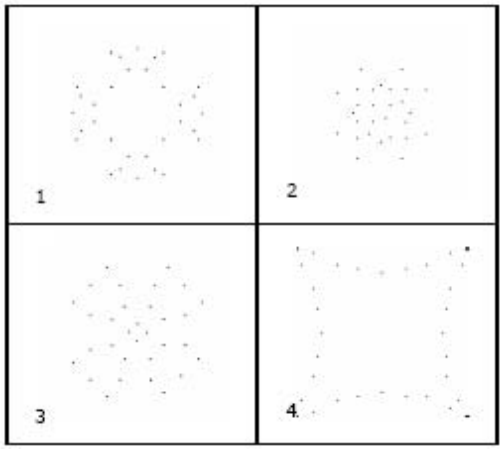


Figura 10. Captura de pantallas para la simulación "Same Point Same Time"

Una vez más, los resultados no arrojan siniestros registrados. Sin embargo, el índice de incidentes conseguido fue superior a las experiencias de Hill. Esto se debe, principalmente, al método de resolución de la problemática. Como se explicó recién, en aquella simulación, los aviones, para sortear la posibilidad de incurrir en colisiones frontales o llegar a destino por una ruta lo más directa posible, generan "revoluciones" u "olas" haciendo que la eficiencia del algoritmo baje notablemente. En nuestro caso, los aviones al intentar seguir una ruta lo más parecida a la ideal, y sólo maniobrando para evitar violar la "zona de exclusión" de otro avión, los retrasos por maniobras caen de forma pronunciada aumentando visiblemente la eficiencia.

7.4 Escenario: Aeropuerto

En un escenario real, los aviones trabajan uniendo destinos distantes. Cada avión en un instante de tiempo T tiene un destino que alcanzar. En su progresión el avión será responsable de elegir la ruta más eficiente. En su camino sólo tendrá un par de restricciones: otros aviones en vuelo; y la capacidad del aeropuerto en el que debe aterrizar. En espacio aéreo abierto, el avión deberá resolver los conflictos evitando violar las zonas de exclusión de otros aviones. Al intentar aterrizar el avión deberá tener en cuenta: no interrumpir otro avión en maniobras de aterrizaje o despegue; y verificar que la capacidad del aeropuerto no esté comprometida. En caso de que el avión no pueda llevar adelante el aterrizaje, queda en espera realizando circuitos de espera en las cercanías del aeropuerto hasta que el mismo quede libre.

8. CONCLUSIÓN Y FUTURAS LÍNEAS DE INVESTIGACIÓN

La capacidad de coordinación y de cooperación de los agentes para cumplir con los objetivos facilita una aproximación de un modelo multiagentes bajo una arquitectura basada en creencias, deseos e intenciones de los agentes que implementan el concepto de Free Flight, los aviones, así, son sus propios controladores. Pueden elegir sus propias rutas de navegación, velocidad, altura, regimenes de ascenso y descenso, maniobras. Para el apoyo, análisis y gestión de una ambiente de vuelo libre dada

sus características propias de un sistema descentralizado sin un director central a cargo del control aéreo, se considero la caracterización de un modelo de simulación basado en un entorno de vida artificial a fin de observar la evolución del espacio aéreo bajo un concepto de Free Flight, para una población inicial de aeronaves con intenciones diversas, respecto a sus aeropuertos de destino. Dentro de las futuras líneas de investigación se considera la elaboración de un Framework para correr el modelo multiagente destinado al apoyo, análisis y gestión de una ambiente de vuelo libre, y la elaboración de ontologías para el concepto de Free Flight

REFERENCIAS

[1] Ronald E.G. Davies. http://usinfo.state.gov/ journals/ites/1000/ijes/trans6.htm
[2] Boeing Company. http://www.boeing.com/
[3] Federal Aviation Administration. http://www.faa.gov/
[4] J2SE. Java 2 Platform, Standard Edition. http://java.sun.com/javase/
[5] NASA. Free Flight Simulation. http://human-factors.arc.nasa.gov /cognition/research/freeflgt.html
[6] The OASIS Air Traffic Management System.
[7] Air Traffic Control (ATC). Aspects of Free Flight. http://www.freeflightatm.org/,
[8] Free Flight with airbone separation assurance. http:/ /www2.nlr.nl/public/hosted-sites/freeflight/
[9] Knowledge Query and Manipulation Language, KQML. http://www.cs.umbc.edu/kqml
[10] FIPA. Foundation for Intelligent Physical Agents. http://www.fipa.org/
[11] J2EE. Java 2 Platform, Enterprise Edition. http://java.sun.com/javaee/
[12] J2ME. Java 2 Platform, Micro Edition. http://java.sun.com/javame/
[13] The Foundation for Intelligent Physical Agents. http://www.fipa.org/, visitado en junio de 2006
[14] Anand S. Rao, Michael P. Georgeoff. BDI Agents: From Theory to Practice. http://www.agent.ai/ doc/upload/200302/rao95.pdf
[15] Aeronautics Learning Laboratory http://www. allstar.fiu.edu/AERO/TCAS.htm
[16] Adami, C, Introducción to Artificial Life,Springer-Verlag, Nueva Cork,1998
[17] JADE. Java Agent DEvelopment Framework. http://jade.tilab.com/
[18] Proyecto Protegè. http://protege.stanford.edu/
[19] MaSE. Multiagent Systems Engineering. http://macr.cis.ksu.edu/projects/mase.htm
[20] StarLogo on the Web. http://education.mit. edu/starlogo
[21] Jared et al A cooperative multi-agent approach to free flight, 2005
[22] J. Krozel, M. Peters, K. D. Bilimoria, C. Lee, and J. S. B. Mitchell, “System performance characteristics of centralized and decentralized air traffic separation strategies,” Fourth USA/Europe Air Traffic Management Research and Development Seminar, 2001.
[23] D. Dugail, E. Feron, and K. Bilimoria, “Stability of intersecting aircraft flows using heading change maneuvers for conflict avoidance,” American Control Conference, 2002.
[24] Z.-H. Mao, E. Feron, and K. Bilimoria, “Stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules,” IEEE Transactions on Intelligent Transporation Systems, vol. 2, pp. 101–109, 2001.
[25] J.C. Hill, F.R Johnson, J.M. Archibald, R.L. Frost, W.C. Stirling, “A Cooperative Multi-Agent Approach to Free Flight”, AAMAS’05, Julio 25-29, 2005, Utrecht, Netherlands

Defeasible Decision Making in a Robotic Environment

Edgardo Ferretti, Marcelo Errecalde
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)*
Departamento de Informática, Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
(D5700HHW) - San Luis - Argentina
Tel: (02652) 420823 / Fax: (02652) 430224
e-mail: {ferretti, merreca}@unsl.edu.ar

and

Alejandro García,‡ Guillermo Simari
‡ Consejo Nacional de Investigaciones Científicas y Técnicas
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)†
Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur
Av. Alem 1253, (B8000CPB) Bahía Blanca, Argentina
Tel: (0291) 459-5135 / Fax: (0291) 459-5136
e-mail: {ajg, grs}@cs.uns.edu.ar

Abstract

Decision making models for autonomous agents are recently receiving increased attention, particularly in the field of intelligent robots. This work presents a Defeasible Logic Programming approach to decision making in an environment with single and multiple robots. We will show, how a successful tool for knowledge representation and defeasible reasoning could be applied to the problem of deciding which task should be performed next. Besides, we will explain with detailed examples how the decision process is performed when there is only one robot in the environment, and then we will consider how the same robot decides when there are more robots working in the environment.

Keywords: Cognitive Robotics, Decision making, DeLP, Khepera robot, Webots.

Resumen

Actualmente, los modelos de toma de decisiones para agentes autónomos están recibiendo mucha atención, particularmente en el área de robots inteligentes. Este trabajo presenta un enfoque basado en Programación en Lógica Rebatible para la toma de decisiones en un ambiente con un único robot y con múltiples robots. Mostraremos como una herramienta exitosa para la representación de conocimiento y razonamiento rebatible, puede ser aplicada al problema de decidir que tarea debe ser realizada a continuación. Además, explicaremos con ejemplos detallados como se realiza el proceso de decisión cuando hay solamente un robot en el ambiente, y luego consideraremos como decide el mismo robot cuando hay otros robots presentes en el ambiente.

Palabras claves: Robótica Cognitiva, Toma de decisiones, DeLP, robot Khepera, Webots.

*Partially supported by Universidad Nacional de San Luis and the ANPCyT (PICT 2002, Nro. 12600).
†Partially supported by Universidad Nacional del Sur, the ANPCyT (PICT 2002, Nro. 13096) and CONICET (PIP 5050).

1 INTRODUCTION

Decision making models for autonomous agents have received increased attention, particularly in the field of intelligent robots. The proposed models are often based on formal theories of decision, such as classical Decision Theory [13], Qualitative Decision Theory [6] and BDI logics [22]. In other cases, models from neuroscience, cognitive psychology and ethology are considered. In these models, the agents' decision making process is an emergent phenomenon of the interaction of elemental behaviors [5]. An established approach to decision making in robotic systems is that of reactive decision systems. In such systems it is assumed that the agent's success is determined by its capacity to appropriately react solely to external stimuli. The decision process is thus usually dedicated to the selection of the action to be executed, based on the current perceptual information with little (if any) pre-processing.

When applicable, the reactive approach has the advantages of simplicity and speed. However, there are domains in which this approach to decision making becomes exceedingly difficult to apply or may not intuitively describe the behavior desired. In such cases, the agent's decision can be partly determined by immediate perceptual data but may also include a complete history of previous perceptions and decisions. The agent may also need to consider questions such as: *Which is the more appropriate goal to pursue in the current situation? Which one of the alternative plans do I have to select to reach a certain goal? Can I carry out this task on my own or I should request help from another agent?* Further complicating matters, the information used in the decision process is (in most domains) incomplete and potentially inconsistent.

To address these issues, we propose the use of a defeasible argumentation formalism. In this paper we will show how a Defeasible Logic Programming approach could be applied in a robotic domain for knowledge representation and reasoning about which task to perform next. At this end, we have selected a simple application domain, consisting of different scenarios where simulated robots perform cleaning tasks. We use the professional simulator (see Figure 1) *Webots* [18], to simulate the *Khepera* 2 robots [14]. The *Khepera* 2 robot is a miniature mobile robot which has eight infrared sensors to measure ambient light levels and proximity to nearby objects. Two independent variable speed motors enable forward and backward motion as well as turns at different speeds. A gripper-arm extension module allows the handling of small objects.

The environment (see Figure 1) consists of a square arena of 100 units per side which is conceptually divided into square cells of 10 units per side each. In this environment, more than one robot could be acting at the same time (Figure 1(c)), but there is no communication among them. There is a global camera which provides the necessary information to perform their activities. The *store* is a 30×30 units square on the top-right corner and represents the target area where the boxes should be transported. There are boxes of three different sizes (*small*, *medium* and *big*) spread over the environment.

Due to physical constraints on the capabilities of the gripper-arms, the robots can grab small and medium boxes, but because of their size the big ones cannot be handled. Nevertheless, a robot is able to move a big box by pushing it. At most two boxes can be stacked, but a box cannot be stacked on top of a smaller box. Therefore, big boxes are always on the floor.

The autonomy of the robots is limited and they cannot measure the state of their batteries, thus, the robots could not perform a globally optimized task. Because of this drawback, a *greedy* strategy is used to select the next box. To perform the reasoning, a robot will use the perceptual information about the boxes and other robots, and its preferences which will be represented with defeasible rules. For example, a robot could prefer the smallest box, or the

nearest one, or the box that is nearest to the store. As we will show below, these preferences will be defeasible and they may change according to the current situation or the presence of other robots. Arguments for and against selecting a box will be considered in order to select the more appropriate one.

A robot capable of solving this kind of problems must at least address the following issues: to perceive the surrounding world, to decide which goal has to be reached and to have the capabilities for reaching this goal. Several architectures have been proposed in the literature which provide the agents with these skills [12, 7, 23]. In this work, we only consider the necessary reasoning processes to make decisions about which is the most suitable box to be transported by the robots. We will not address low-level aspects related to sensorial perception, like image acquisition and processing. Besides, we are not going to consider problems related to the implementation of low-level actions, like object handling and the robots' navigation system. Some of the aspects related to the sensorial and effectorial support for the Khepera robots have been presented elsewhere [8, 9].

The paper is organized as follows. In Section 2 we explain how robots represent their knowledge and include an overview of the reasoning formalism they use. Section 3 shows how the decision process is performed when there is only one robot in the environment. Then, in Section 4 we show how the robot decides when there are more robots working in the environment. In Section 5 related work is described. Finally, Section 6 offers some conclusions.

(a) (b) (c)

Figure 1: Three different scenarios

2 KNOWLEDGE REPRESENTATION AND REASONING

Figure 1(b) shows an example where there is only one robot ($khep1$) and four boxes: two small ($box1$ and $box2$) which are at the same distance from the robot, $box3$ that is medium size and is near to the store, and $box4$ that is big and is far from both, robot and store. Considering its preferences the robot will obtain arguments for and against selecting each box.

For example, there is an argument *for* selecting $box1$ because is near to the robot and is small, but there is an argument *against* selecting $box1$ (counter-argument) because there is another small box near to the robot ($box2$) that is nearer to the store than $box1$. As will be shown below a dialectical analysis involving arguments and counter-arguments will be performed to decide which argument prevails. In this case the box chosen will be $box2$, because it is small, and is nearer to the store. Since the environment is dynamic, if something changes, then, new arguments can be generated or other can be invalidated. Thus, the selected box may be different. For instance, let us consider Figure 1(c), that differs from Figure 1(b) in that there is

one more robot ($khep2$) in the environment. Here, using this new information, the robot $khep1$ will choose $box1$ because it has a new argument against selecting $box2$:"there is another robot ($khep2$) that will choose $box2$ on the grounds that it is its nearest smaller box." In this way, the overall performance is enhanced avoiding a conflict in the robots' selection choices.

The robot's knowledge about the environment and its preferences for selecting a box will be represented using Defeasible Logic Programming (DeLP) [10] a formalism that combines logic programming and defeasible argumentation.

In DeLP, knowledge is represented with a program $\mathcal{P}$ that contains facts, strict rules and defeasible rules. *Facts* are ground literals representing atomic information or the negation of atomic information. In our application examples, facts will be used for representing perceptual information about the environment, (*e.g.*, $box(box2)$ or $near(box2, store)$). *Strict Rules*, are denoted $L_0 \leftarrow L_1, \ldots, L_n$, where the *head* L_0 is a ground literal and the *body* $\{L_i\}_{i>0}$ is a set of ground literals. These kind of rules will be used for representing firm information (*e.g.*, $\sim near(box1, khep1) \leftarrow far(box1, khep1)$). On the other hand, *Defeasible Rules*, denoted $L_0 \prec L_1, \ldots, L_n$, where the *head* L_0 is a ground literal and the *body* $\{L_i\}_{i>0}$ is a set of ground literals, represent tentative reasons for (or against) selecting a box (*e.g.*, $choose(X) \prec small(X)$, or $\sim choose(X) \prec small(X), far(X, khep1)$).

Syntactically, the symbol "$\prec$" is all that distinguishes a defeasible rule from a strict one. Pragmatically, a defeasible rule is used to represent defeasible knowledge, *i.e.*, tentative information that may be used if nothing could be posed against it. A defeasible rule "$Head \prec Body$" is understood as expressing that "*reasons to believe in the antecedent Body provide reasons to believe in the consequent Head*" [24]. When required, a Defeasible Logic Program $\mathcal{P}$ is denoted (Π, Δ) where $\Pi{=}\Pi_f \cup \Pi_r$, distinguishing the subset Π_f of facts, strict rules Π_r, and the subset Δ of defeasible rules. Observe that strict and defeasible rules are ground. However, following the usual convention [16], some examples will use "schematic rules" with variables. Given a "schematic rule" R, $Ground(R)$ stands for the set of all ground instances of R. Given a program $\mathcal{P}$ with schematic rules, we define: $Ground(\mathcal{P}) = \bigcup_{R \in \mathcal{P}} Ground(R)$. In order to distinguish variables, they are denoted with an initial uppercase letter.

Strong negation is allowed in the head of program rules, and hence may be used to represent contradictory knowledge. From a program (Π, Δ) contradictory literals could be derived, however, the set Π (which is used to represent non-defeasible information) must possess certain internal coherence. Therefore, Π has to be non-contradictory, *i.e.*, no pair of contradictory literals can be derived from Π. Given a literal L the complement with respect to strong negation will be denoted $\overline{L}$ (*i.e.*, $\overline{a}{=}\sim a$ and $\overline{\sim a}{=}a$.)

To deal with contradictory and dynamic information, in DeLP, *arguments* for conflicting pieces of information are built and then compared in order to decide which one prevails. The argument that prevails provides a *warrant* for the information that it supports. A brief explanation of how warrants are obtained using DeLP is included below (the interested reader is referred to [10] for a detailed explanation.[1])

In DeLP a literal L is *warranted* from (Π, Δ) if a non-defeated argument $\mathcal{A}$ supporting L exists. To put it briefly, an *argument* for a literal L, denoted $\langle \mathcal{A}, L \rangle$, is a minimal set of defeasible rules $\mathcal{A}{\subseteq}\Delta$, such that $\mathcal{A} \cup \Pi$ is non-contradictory and there is a derivation for L from $\mathcal{A} \cup \Pi$. To establish if $\langle \mathcal{A}, L \rangle$ is non-defeated, *argument rebuttals* or *counter-arguments* that could be *defeaters* for $\langle \mathcal{A}, L \rangle$ are considered, *i.e.*, counter-arguments that by some criterion are preferred to $\langle \mathcal{A}, L \rangle$. Since counter-arguments are arguments, defeaters for them may exist, and defeaters

[1] The implementation (interpreter) of DeLP that satisfies the semantics described in [10] is currently accessible online at `http://lidia.cs.uns.edu.ar/DeLP`.

for these defeaters, and so on. Thus, a sequence of arguments called *argumentation line* is constructed, where each argument defeats its predecessor in the line (for a detailed explanation of this dialectical process see [10].) In DeLP, given a query Q there are four possible answers: YES, if Q is warranted; NO, if the complement of Q is warranted; UNDECIDED, if neither Q nor its complement are warranted; and UNKNOWN, if Q is not in the language of the program.

3 SINGLE ROBOT BOX SELECTION

In this section we describe the processes involved in deciding which box to transport when there is only one robot in the environment.

Example 1 *Consider the simple scenario depicted in Figure 1(a) where there is a single robot (khep1), and three boxes: box1 (small) and box4 (big) near to the robot, and box3 (medium) near to the store. The knowledge of the robot, referring to this particular scenario, will be represented with the defeasible logic program* $\mathcal{P}_1 = (\Pi_1, \Delta_1)$ *shown in Figures 2 and 3.*

The set Π_1 contains a subset Π_{f1} of facts representing the perception of the current situation (as shown in Figure 2(a)) and a subset Π_{r1} of strict rules shown in Figure 2(b). The facts of Π_{f1} are obtained from perception functions (see [8]) and represent information about the objects present in the environment (facts (1)-(8)) and the position of the objects (facts (9)-(15)). The strict rules in Figure 2(b) represent non-defeasible information, for example, rule (16) states that if X is far from an object O (in this case O can be instantiated with, $khep1$ or $store$) then it is not near to O. Rules (17)-(24) define size relationships among the boxes. In rule (25) is stated that an object X is nearer than an object Y with respect to an object O, if X is near to O and Y is not. Besides, rules (26) and (27) define strict reasons for choosing a box X. For instance, if X is the last box in the environment this is a firm reason to be chosen (rule (26)) but if X has a box on its top, this is a negative reason for choosing it (rule (27)).

$robot(khep1)$	(1)	$on(box1, box4)$	(9)	$\sim near(X,O) \leftarrow far(X,O)$	(16)
$self(khep1)$	(2)	$near(box1, khep1)$	(10)	$same\_size(X,Y) \leftarrow small(X), small(Y)$	(17)
$box(box1)$	(3)	$near(box4, khep1)$	(11)	$same\_size(X,Y) \leftarrow medium(X), medium(Y)$	(18)
$box(box3)$	(4)	$near(box3, store)$	(12)	$same\_size(X,Y) \leftarrow big(X), big(Y)$	(19)
$box(box4)$	(5)	$far(box1, store)$	(13)	$smaller(X,Y) \leftarrow small(X), medium(Y)$	(20)
$small(box1)$	(6)	$far(box3, khep1)$	(14)	$smaller(X,Y) \leftarrow small(X), big(Y)$	(21)
$medium(box3)$	(7)	$far(box4, store)$	(15)	$smaller(X,Y) \leftarrow medium(X), big(Y)$	(22)
$big(box4)$	(8)			$\sim smaller(X,Y) \leftarrow same\_size(X,Y)$	(23)
				$\sim smaller(X,Y) \leftarrow smaller(Y,X)$	(24)
				$nearer\_than(X,Y,O) \leftarrow near(X,O), \sim near(Y,O)$	(25)
				$choose(X) \leftarrow unique(X)$	(26)
				$\sim choose(X) \leftarrow on(Y,X)$	(27)

(a) Π_{f1} (b) Π_{r1}

Figure 2: Facts and strict rules of Π_1

Figure 3 includes the defeasible rules of Δ_1. Rules (28)-(30) provide defeasible reasons to determine if a box X is smaller than a box Y when strict rules (17)-(24) cannot be used, because of the lack of information about the boxes' size. In the same way, rules (31)-(35) provide evidence to the determine if a box X is nearer to the robot (or to the store), than a box Y if there is no enough information about the proximity of these boxes to the robot (or to the store). If enough information is available the strict rule (25) can be used instead. Furthermore, rules (36)-(51) model preference criteria with respect to the size and location of the boxes. For example, rules (36) and (37) represent the preferences of the robot with respect

to the boxes' size, the smaller ones are preferred. Moreover, rules (40) and (44) show that being near to the store or to the robot has a higher priority than the boxes' size. In addition, rule (48) and (50) state that boxes near to the robot are more desirable than those near to the store, independently of their size. Rules (52) and (53) represent the (defeasible) criteria used by the robot to choose the box to be transported. These rules, differ from rules (26) and (27) in that they are based on the comparison between the features of two boxes. Thus, the rules to determine if the robot prefers a box X over a box Y ($pref(X,Y)$) play the role of a rational preference relation, as defined in classical decision theory [17]. The main difference of our approach lays in that the preferences are defeasible, they are based on an argumentation system that manages incomplete information about the environment, and in consequence they can change when new information is available.

$$\sim smaller(X,Y) \prec box(X), box(Y) \quad (28)$$
$$smaller(X,Y) \prec box(X), box(Y), small(X) \quad (29)$$
$$smaller(X,Y) \prec box(X), box(Y), big(Y) \quad (30)$$
$$\sim nearer\_than(X,Y,O) \prec box(X), box(Y) \quad (31)$$
$$nearer\_than(X,Y,O) \prec box(X), box(Y), near(X,O) \quad (32)$$
$$\sim nearer\_than(X,Y,O) \prec box(X), box(Y), \sim near(X,O) \quad (33)$$
$$\sim nearer\_than(X,Y,O) \prec box(X), box(Y), near(X,O), near(Y,O) \quad (34)$$
$$\sim nearer\_than(X,Y,O) \prec box(X), box(Y), \sim near(X,O), \sim near(Y,O) \quad (35)$$
$$pref(X,Y) \prec smaller(X,Y) \quad (36)$$
$$\sim pref(X,Y) \prec \sim smaller(X,Y) \quad (37)$$
$$pref(X,Y) \prec nearer\_than(X,Y,store), smaller(X,Y) \quad (38)$$
$$\sim pref(X,Y) \prec nearer\_than(Y,X,store), \sim smaller(X,Y) \quad (39)$$
$$pref(X,Y) \prec nearer\_than(X,Y,store), \sim smaller(X,Y) \quad (40)$$
$$\sim pref(X,Y) \prec nearer\_than(Y,X,store), smaller(X,Y) \quad (41)$$
$$pref(X,Y) \prec self(Z), nearer\_than(X,Y,Z), smaller(X,Y) \quad (42)$$
$$\sim pref(X,Y) \prec self(Z), nearer\_than(Y,X,Z), smaller(X,Y) \quad (43)$$
$$pref(X,Y) \prec self(Z), nearer\_than(X,Y,Z), \sim smaller(X,Y) \quad (44)$$
$$\sim pref(X,Y) \prec self(Z), nearer\_than(Y,X,Z), \sim smaller(X,Y) \quad (45)$$
$$pref(X,Y) \prec self(Z), nearer\_than(X,Y,Z), nearer\_than(X,Y,store), smaller(X,Y) \quad (46)$$
$$pref(X,Y) \prec self(Z), nearer\_than(X,Y,Z), nearer\_than(X,Y,store), \sim smaller(X,Y) \quad (47)$$
$$\sim pref(X,Y) \prec self(Z), nearer\_than(Y,X,Z), nearer\_than(X,Y,store), smaller(X,Y) \quad (48)$$
$$pref(X,Y) \prec self(Z), nearer\_than(X,Y,Z), nearer\_than(Y,X,store), smaller(X,Y) \quad (49)$$
$$\sim pref(X,Y) \prec self(Z), nearer\_than(Y,X,Z), nearer\_than(X,Y,store), \sim smaller(X,Y) \quad (50)$$
$$pref(X,Y) \prec self(Z), nearer\_than(X,Y,Z), nearer\_than(Y,X,store), \sim smaller(X,Y) \quad (51)$$
$$choose(X) \prec diff(X,Y), pref(X,Y) \quad (52)$$
$$\sim choose(X) \prec diff(X,Y), \sim pref(X,Y) \quad (53)$$

Figure 3: Defeasible rules of Δ_1

In the situation described in Example 1, it is clear that the robot should choose the small box ($box1$) because it is near to itself, it should not choose $box4$ because $box1$ is on its top and it should not choose $box3$ because there is a smaller box that can be chosen. From $\mathcal{P}_1$ there are four arguments supporting $choose(box1)$:

$$\mathcal{A}_1 = \left\{ \begin{array}{l} choose(box1) \prec diff(box1,box3), pref(box1,box3) \\ pref(box1,box3) \prec smaller(box1,box3) \end{array} \right\}$$

$$\mathcal{A}_2 = \left\{ \begin{array}{l} choose(box1) \prec diff(box1,box3), pref(box1,box3) \\ pref(box1,box3) \prec self(khep1), \\ \qquad nearer\_than(box1,box3,khep1), \\ \qquad smaller(box1,box3) \end{array} \right\}$$

$$\mathcal{A}_3 = \left\{ \begin{array}{l} choose(box1) \prec diff(box1,box3), pref(box1,box3) \\ pref(box1,box3) \prec self(khep1), \\ \qquad nearer\_than(box1,box3,khep1), \\ \qquad nearer\_than(box3,box1,store), \\ \qquad smaller(box1,box3) \end{array} \right\}$$

$$\mathcal{A}_4 = \left\{ \begin{array}{l} choose(box1) \prec diff(box1,box4), pref(box1,box4) \\ pref(box1,box4) \prec smaller(box1,box4) \end{array} \right\}$$

Since all the arguments $\mathcal{A}_1$-$\mathcal{A}_4$ have no defeaters, then the DeLP answer for $choose(box1)$ is YES. On the other hand, the answers for $choose(box3)$ and $choose(box4)$ are NO.

Example 2 *Consider now the situation presented at the beginning of this paper and depicted in Figure 1(b). There is a single robot (khep1) and two small boxes: box1 near to the robot, and*

box2 near to the robot and to the store. There are also a medium size box (box3) near to the store and a big one (box4) far from both, robot and store. The knowledge of khep1, referring to this particular scenario, will be represented with the defeasible logic program $\mathcal{P}_2 = (\Pi_2, \Delta_2)$*, where the perceptions of the current situation* (Π_{f2}) *are presented in Figure 4(a). Defeasible and strict rules of* $\mathcal{P}_2$ *coincide with the ones of* $\mathcal{P}_1$ *(Example 1), i.e.,* $\Pi_{r2}=\Pi_{r1}$*, and* $\Delta_2=\Delta_1$*.*

$robot(khep1)$	(54)	$big(box4)$	(63)
$self(khep1)$	(55)	$near(box1, khep1)$	(64)
$box(box1)$	(56)	$near(box2, khep1)$	(65)
$box(box2)$	(57)	$near(box2, store)$	(66)
$box(box3)$	(58)	$near(box3, store)$	(67)
$box(box4)$	(59)	$far(box1, store)$	(68)
$small(box1)$	(60)	$far(box3, khep1)$	(69)
$small(box2)$	(61)	$far(box4, khep1)$	(70)
$medium(box3)$	(62)	$far(box4, store)$	(71)

(a) From figure 1(b)

$robot(khep1)$	(72)	$near(box1, khep1)$	(83)
$robot(khep2)$	(73)	$near(box2, khep1)$	(84)
$self(khep1)$	(74)	$near(box2, khep2)$	(85)
$box(box1)$	(75)	$near(box2, store)$	(86)
$box(box2)$	(76)	$near(box3, store)$	(87)
$box(box3)$	(77)	$near(box4, khep2)$	(88)
$box(box4)$	(78)	$far(box1, store)$	(89)
$small(box1)$	(79)	$far(box1, khep2)$	(90)
$small(box2)$	(80)	$far(box3, khep1)$	(91)
$medium(box3)$	(81)	$far(box3, khep2)$	(92)
$big(box4)$	(82)	$far(box4, khep1)$	(93)
		$far(box4, store)$	(94)

(b) From figure 1(c)

Figure 4: $khep1's$ perceptual information

In the situation described in Example 2, it is evident that the robot should choose $box2$ because is near to itself and is also near to the store. From the program $\mathcal{P}_2$ there are seven arguments ($\mathcal{A}_5$-$\mathcal{A}_{11}$) supporting $choose(box2)$:

$$\mathcal{A}_5 = \left\langle \begin{array}{l} choose(box2) \prec diff(box2, box1), pref(box2, box1) \\ pref(box2, box1) \prec nearer\_than(box2, box1, store), \\ \quad \sim smaller(box2, box1) \end{array} \right\rangle \quad \mathcal{A}_6 = \left\langle \begin{array}{l} choose(box2) \prec diff(box2, box3), pref(box2, box3) \\ pref(box2, box3) \prec smaller(box2, box3) \end{array} \right\rangle$$

$$\mathcal{A}_7 = \left\langle \begin{array}{l} choose(box2) \prec diff(box2, box3), pref(box2, box3) \\ pref(box2, box3) \prec self(khep1) \\ \quad nearer\_than(box2, box3, khep1), \\ \quad smaller(box2, box3) \end{array} \right\rangle \quad \mathcal{A}_8 = \left\langle \begin{array}{l} choose(box2) \prec diff(box2, box4), pref(box2, box4) \\ pref(box2, box4) \prec smaller(box2, box4) \end{array} \right\rangle$$

$$\mathcal{A}_9 = \left\langle \begin{array}{l} choose(box2) \prec diff(box2, box4), pref(box2, box4) \\ pref(box2, box4) \prec nearer\_than(box2, box4, store), smaller(box2, box4) \end{array} \right\rangle$$

$$\mathcal{A}_{10} = \left\langle \begin{array}{l} choose(box2) \prec diff(box2, box4), pref(box2, box4) \\ pref(box2, box4) \prec self(khep1), nearer\_than(box2, box4, khep1), smaller(box2, box4) \end{array} \right\rangle$$

$$\mathcal{A}_{11} = \left\langle \begin{array}{l} choose(box2) \prec diff(box2, box4), pref(box2, box4) \\ pref(box2, box4) \prec self(khep1), nearer\_than(box2, box4, khep1), nearer\_than(box2, box4, store), smaller(box2, box4) \end{array} \right\rangle$$

Finally, from $\mathcal{P}_2$, the answers for $choose(box1)$, $choose(box3)$ and $choose(box4)$ are NO.

4 REASONING ABOUT ROBOTS

This section describes how a robot can decide when there are more robots working in the same environment. That is, the robot has to reason about other robots' choices. This topic must be considered in the reasoning processes of the robot because the presence of other robots in the environment require a coordinated behavior among them. Let us consider the following example to see how to implement this issue in a direct way in a DeLP-program.

Example 3 *Consider the situation shown in Figure 1(c). This situation extends the one presented in Figure 1(b) in that there is a second robot (khep2) in the environment. The knowledge of the robots, referring to this particular scenario, will be represented with the DeLP-programs* $\mathcal{P}_{3.1} = (\Pi_{3.1}, \Delta_3)$ *for khep1 and* $\mathcal{P}_{3.2} = (\Pi_{3.2}, \Delta_3)$ *for khep2. The perception of*

the current situation for khep1 ($\Pi_{f3.1}$) *is presented in Figure 4(b). The perception of khep2,* $\Pi_{f3.2} = \Pi_{f3.1} - \{self(khep1)\} \cup \{self(khep2)\}$. *The defeasible and strict rules of* $\mathcal{P}_{3.1}$ *and* $\mathcal{P}_{3.2}$ *coincide with the ones of* $\mathcal{P}_2$ *(Example 2), i.e.,* $\Pi_{r3.1}$=$\Pi_{r3.2}$=Π_{r2}, *and* Δ_3=Δ_2.

Up to this point, *khep*1 is not able of modelling *khep*2′*s* preferences because all the rules in $\mathcal{P}_{3.1}$ take into account only *khep*1′*s* preferences. However, if we include in $\mathcal{P}_{3.1}$ the rules presented in Figure 5, *khep*1 is now able of considering *khep*2′*s* preferences. Rule (95) allows *khep*1 to determine if there are more robots in the environment. If there are more robots in the environment (in particular there is only one in this situation), with the predicate $choose\_other(R, X)$, *khep*1 is able of determining which box X will be selected by the robot R. To put it briefly, $choose\_other(khep2, X)$ calls the DeLP interpreter from *khep*1 with $\mathcal{P}_{3.2}$ and receives in X the box that *khep*2 will choose. In this way, using rules (96)-(99), *khep*1 is able to select a different box to the one chosen by the other robots, considering in addition its own preferences. The rules shown in Figure 5 are strict because the fact that a box X was chosen by other robot is a non-defeasible reason for not choosing X.

$other(R) \leftarrow robot(R), self(Z), diff(R, Z)$ (95)
$select\_ot(X) \leftarrow other(R), choose\_other(R, X)$ (96)
$\sim select\_ot(X) \leftarrow select\_ot(Y), diff(X, Y)$ (97)
$choose(X) \leftarrow diff(X, Y), pref(X, Y), \sim select\_ot(X)$ (98)
$\sim choose(X) \leftarrow diff(X, Y), pref(X, Y), select\_ot(X)$ (99)

Figure 5: Rules to consider other robot preferences

As it can be noted, *khep*1′*s* choice changes with respect to the selection made by *khep*1 in Example 2. Now, *khep*1 has additional information about the presence of other robot in the environment and the new rules presented in Figure 5 allow it to choose a different box. In this case, *khep*1 will choose *box*1 on the grounds that it beliefs that *khep*2 will choose *box*2, because it is its nearest smaller box near to the store, too. In this way, the overall performance is enhanced avoiding a conflict in the robots' selection choices. If we do not include in $\mathcal{P}_{3.1}$ rules to model other robots' preferences, both *khep*1 and *khep*2 will choose *box*2 as the box to be transported, because it is their smaller box near to themselves and near to the store. It is not possible to foresee which robot will transport the box, if any, because both can grab the box at the same time leading to an unexpected situation.

Let us consider the above-mentioned in more detail. From the knowledge of *khep*1, represented by the DeLP-program $\mathcal{P}_{3.1} = (\Pi_{3.1}, \Delta_3)$, the answer for $choose(box2)$ is NO because there are seven non-defeated arguments $\mathcal{A}_{12}$-$\mathcal{A}_{18}$ supporting $\sim choose(box2)$. To derive $\sim choose(box2)$, rules (40), (95), (96) and (99) are used in $\mathcal{A}_{12}$, but rules (95), (96) and (99) do not appear in $\mathcal{A}_{12}$ because they are strict. The same occurs with $\mathcal{A}_{13}$-$\mathcal{A}_{18}$, all of them use rules (95), (96) and (99) but only the defeasible rules are shown in these arguments. In some arguments the defeasible rules used to derive the preference between two different boxes, are the same, *e.g.*, $\mathcal{A}_{13}$ and $\mathcal{A}_{15}$ use rule (36), while in other cases, different defeasible rules are used to determine the same preference between two boxes *e.g.*, $\mathcal{A}_{13}$ and $\mathcal{A}_{14}$.

$$\mathcal{A}_{12} = \{\ pref(box2, box1) \prec nearer\_than(box2, box1, store), \sim smaller(box2, box1)\ \}$$

$$\mathcal{A}_{13} = \{\ pref(box2, box3) \prec smaller(box2, box3)\ \}$$

$$\mathcal{A}_{14} = \{\ pref(box2, box3) \prec self(khep1), nearer\_than(box2, box3, khep1), smaller(box2, box3)\ \}$$

$$\mathcal{A}_{15} = \{\ pref(box2, box4) \prec smaller(box2, box4)\ \}$$

$$\mathcal{A}_{16} = \{\ pref(box2, box4) \prec nearer\_than(box2, box4, store), smaller(box2, box4)\ \}$$

$$\mathcal{A}_{17} = \{\ pref(box2, box4) \prec self(khep1), nearer\_than(box2, box4, khep1), smaller(box2, box4)\ \}$$

$$\mathcal{A}_{18} = \{\ pref(box2, box4) \prec self(khep1), nearer\_than(box2, box4, khep1), nearer\_than(box2, box4, store), smaller(box2, box4)\ \}$$

Besides, the answer for $choose(box1)$ is YES because there are two non-defeated arguments($\mathcal{A}_{26}$ and $\mathcal{A}_{27}$) supporting this query. It is important to note, that arguments $\mathcal{A}_{26}$ and $\mathcal{A}_{27}$ are obtained using rules (36), (96)-(98) but rule (36) is the only one appearing in the arguments because the other ones are strict. Although, arguments $\mathcal{A}_{21}$-$\mathcal{A}_{25}$ also support $choose(box1)$ they have two blocking defeaters ($\mathcal{A}_{19}$ and $\mathcal{A}_{20}$) supporting $\sim choose(box1)$. The argumentation lines for the query $choose(box1)$ are shown in Figure 6. (Black triangles represent defeated arguments, white triangles non-defeated ones and dotted arrows the blocking defeat relation.) Furthermore, the answers for $choose(box3)$ and $choose(box4)$ are NO, as expected.

$$\mathcal{A}_{19} = \left\{ \begin{array}{l} \sim choose(box1) \prec diff(box1, box2), \sim pref(box1, box2) \\ \sim pref(box1, box2) \prec \sim smaller(box1, box2) \end{array} \right\}$$

$$\mathcal{A}_{20} = \left\{ \begin{array}{l} \sim choose(box1) \prec diff(box1, box2), \sim pref(box1, box2) \\ \sim pref(box1, box2) \prec nearer\_than(box2, box1, store), \sim smaller(box1, box2) \end{array} \right\}$$

$$\mathcal{A}_{21} = \left\{ \begin{array}{l} choose(box1) \prec diff(box1, box3), pref(box1, box3) \\ pref(box1, box3) \prec smaller(box1, box3) \end{array} \right\}$$

$$\mathcal{A}_{22} = \left\{ \begin{array}{l} choose(box1) \prec diff(box1, box3), pref(box1, box3) \\ pref(box1, box3) \prec self(khep1), nearer\_than(box1, box3, khep1), smaller(box1, box3) \end{array} \right\}$$

$$\mathcal{A}_{23} = \left\{ \begin{array}{l} choose(box1) \prec diff(box1, box3), pref(box1, box3) \\ pref(box1, box3) \prec self(khep1), nearer\_than(box1, box3, khep1), nearer\_than(box3, box1, store), smaller(box1, box3) \end{array} \right\}$$

$$\mathcal{A}_{24} = \left\{ \begin{array}{l} choose(box1) \prec diff(box1, box4), pref(box1, box4) \\ pref(box1, box4) \prec smaller(box1, box4) \end{array} \right\}$$

$$\mathcal{A}_{25} = \left\{ \begin{array}{l} choose(box1) \prec diff(box1, box4), pref(box1, box4) \\ pref(box1, box4) \prec self(khep1), nearer\_than(box1, box4, khep1), smaller(box1, box4) \end{array} \right\}$$

$$\mathcal{A}_{26} = \{ \; pref(box1, box3) \prec smaller(box1, box3) \; \} \quad \mathcal{A}_{27} = \{ \; pref(box1, box4) \prec smaller(box1, box4) \; \}$$

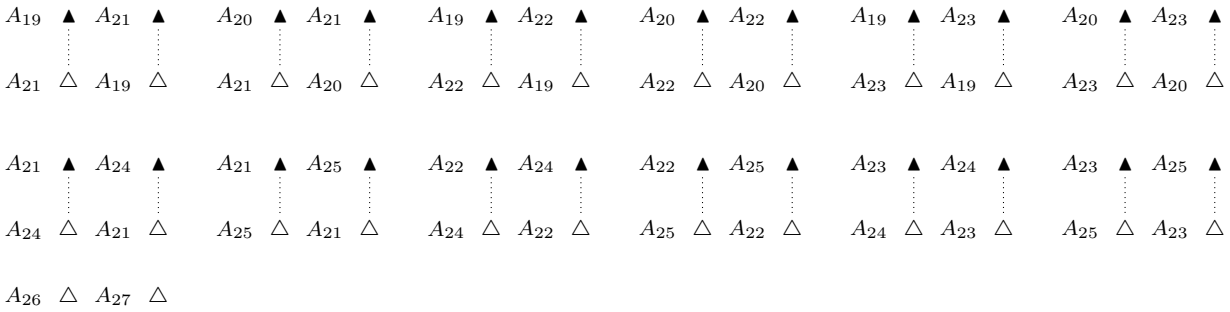


Figure 6: Argumentation lines for $choose(box1)$

Now, if we consider the decision from the $khep2's$ standpoint. From $\mathcal{P}_{3.2} = (\Pi_{3.2}, \Delta_3)$, using rule (46) a non-defeated argument supporting $choose(box2)$ can be built. Moreover, as expected, the answers for $choose(box1)$, $choose(box3)$ and $choose(box4)$ are NO.

It must be pointed out, that the robots coordination is effectively achieved assuming that $khep2$ does not simultaneously take into account $khep1's$ decisions (rules in Figure 5 are not included in $\mathcal{P}_{2.2}$). If this happen, both robots will avoid to choose $box2$ and they will take suboptimal decisions. In a context of recursive modeling this is equivalent to assume that $khep2$ is a 0-level agent that does not recognize the existence of other agents in the world.

5 RELATED WORK

After a proliferation period of a wide variety of reactive robotic architectures [5, 25, 19, 2], it was clear the necessity of introducing high-level deliberative processes in the decision making of autonomous robots. Regardless of the fact that deliberation has many advantages for decision making of an agent, it has the disadvantage of being slow compared to generating actions in a reactive fashion. Consequently, several hybrid architectures that combine the advantages of reactive and goal-directed aspects, were proposed having as their main difference the way they incorporate the deliberative component.

Arkin [1] was among the first to advocate the use of both deliberative and reactive control systems within the autonomous robot architecture, incorporating a traditional planner that could reason over flexible and modular reactive control system. Gat [12] proposed a three-level hybrid system [11] (Atlantis) incorporating a Lisp-based deliberator, a sequencer that handled failures of the reactive system, and a reactive controller. Estlin *et al.* [7] presented a two-layered architecture where the top decision layer contains techniques for autonomy creating a plan of robot commands, and the bottom functional layer provides standard robot capabilities that interface to system hardware. The main attention focus of the above-mentioned works has been the definition of the architecture related components necessary to achieve a successful behavior of the robots in real-life complex problems. Aspects like the description of the reactive component, the support of planning capabilities and the interaction of both components in an adequate planning-execution system for the robots, are considered in detail. Our work adopts a more top-down approach and concentrates on the high-level decision capabilities of the robots, an aspect that, in our opinion, is not sufficiently analyzed in the majority of these works. We consider that problems related to reactive control are important, but more attention should be paid to the deliberative processes involved in the robot's decision making, the main concern of the present paper.

Our proposal is closely related to the approach adopted by Parsons *et al.* [21]. This work incorporates a BDI deliberative component based on the work of Bratman on practical reasoning [4], where the internal state of an agent is determined by its knowledge about the environment (beliefs), the action facilities the agent is able to choose from (desires) and the current goals (intentions). In particular, in our work we follow some of the ideas exposed by Parsons *et al.* about the convenience of integrating high-level reasoning facilities with low-level robust robot control. We share the approach of seeing the low-level module as a black box which receives from the high-level component goals to be achieved, and plans to reach that goals are internally generated, and then an acknowledgement is received to inform failures or if everything finished as planned. Nonetheless, our work has some differences with the proposal of Parsons *et al.* in that we do not use a BDI deliberator as high-level reasoning layer, instead we use a non-monotonic reasoning module based on a defeasible argumentation system.

With respect to this last issue, our approach to decision making is related to other works which use argumentative processes as a fundamental component in the decision making of an agent. In [3], an agent called *Drama* incorporates an argumentation component which provides the ability to make flexible and context dependent decisions about medical treatment, based on several information sources (perspectives). The influence of different contexts that arise in changing environments is also considered in [15] where an argumentation-based framework supports the decision making of an agent modular architecture. In this case, arguments and their strength depend on the particular context that the agent finds himself. The fundamental role of argumentation for the management of uncertainty in symbolic decision making is

highlighted in [20] where several applications based on argumentative approaches are presented as empirical evidence of this claim. It is important to note that these argumentation systems have been usually integrated in *software* agents. Although in this paper we presented simulated scenarios, in our approach defeasible argumentation is applied in a robotic domain where the uncertainty generated by noisy sensors and effectors, changes in the physical environment and incomplete information about it, make this kind of problems a more challenging test-bed for the decision processes of an agent.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have shown how a Defeasible Logic Programming approach could be applied in a robotic domain for knowledge representation and reasoning about which task to perform next. Our approach considers the ability of Defeasible Logic Programming to reason with incomplete and potentially inconsistent information. The simple application domain described consists of different scenarios where simulated robots perform cleaning tasks. We have presented problems and their solutions, when there is only one robot in the environment, and when more than one robot are working in the same environment.

Future work includes considering more complex environments where robots have different sensing and acting capabilities. A deeper study and formalization of the process of implementing preference relations using an argumentative approach is required. Other important issue we are considering, is to extend this work to more general environments, where different levels of modeling of other robots are considered in the argumentation process, as well as their impact in the robots coordination.

REFERENCES

[1] R. C. Arkin. Integrating behavioral, perceptual and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105–122, 1990.

[2] R. C. Arkin. *Behaviour-Based Robotics*. The MIT Press, 1998.

[3] Katie Atkinson, Trevor J. M. Bench-Capon, and Sanjay Modgil. Argumentation for decision support. In *DEXA*, pages 822–831, 2006.

[4] Michael E. Bratman. *Intention, Plans, and Practical Reason*. The David Hume Series. CSLI Publications, Stanford University, 1999.

[5] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[6] Jon Doyle and Richmond H. Thomason. Background to qualitative decision theory. *AI Magazine*, 20(2):55–68, 1999.

[7] T. Estlin, R. Volpe, I. Nesnas, D. Muts, F. Fisher, B. Engelhardt, and S. Chien. Decision-making in a robotic architecture for autonomy. In *International Symposium, on AI, Robotics and Automation for Space*, Montreal, Canada, June 2001.

[8] Edgardo Ferretti, Marcelo Errecalde, Alejandro García, and Guillermo Simari. Khedelp: A framework to support defeasible logic programming for the khepera robots. In *International Symposium on Robotics and Automation*, pages 98–103, 2006.

[9] Edgardo Ferretti, Marcelo Errecalde, Alejandro García, and Guillermo Simari. Khepera robots with argumentative reasoning. In *4th International Symposium on Autonomous Minirobots for Research and Edutainment*, 2-5, October 2007. ACCEPTED.

[10] Alejandro Javier García and Guillermo Ricardo Simari. Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming*, 4(2):95–138, 2004.

[11] E. Gat. Integrating planning and reacting in a heterogeneous asynchronous architecture for mobile robots. *SIGART Bulletin*, 2:70–74, 1991.

[12] E. Gat. On three-layer architectures. In *Artificial Intelligence and Mobile Robots*, 1998.

[13] Richard C. Jeffrey. *The Logic of Decision*. University Of Chicago Press, 2nd edition, 1990.

[14] K-Team. Khepera 2. http://www.k-team.com, 2002.

[15] Antonis Kakas and Pavlos Moraitis. Argumentation based decision making for autonomous agents. In *AAMAS*, pages 883–890, 2003.

[16] V. Lifschitz. Foundations of logic programming. In G. Brewka, editor, *Principles of Knowledge Representation*, pages 69–127. CSLI, 1996.

[17] Andreu Mas-Collel, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[18] Olivier Michel. Webots: Professional mobile robot simulation. *Journal of Advanced Robotics Systems*, 1(1):39–42, 2004.

[19] Stefano Nolfi and Dario Floreano. *Evolutionary Robotics*. The MIT Press, 2000.

[20] Simon Parsons and John Fox. Argumentation and decision making: A position paper. In *FAPR*, pages 705–709, London, UK, 1996. Springer-Verlag.

[21] Simon Parsons, Ola Pettersson, Alessandro Saffiotti, and Michael Wooldridge. *Artificial Intelligence Today: Recent Trends and Developments*, chapter Robots with the Best of Intentions, pages 329–338. Springer, 1999.

[22] A. Rao and M. Georgeff. Decision procedures for bdi logics. *Journal of Logic and Computation*, 8:293–342, 1998.

[23] Nicolás Rotstein, Alejandro García, and Guillermo Simari. Reasoning from desires to intentions: A dialectical framework. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07)*, July 2007. ACCEPTED.

[24] Guillermo R. Simari and Ronald P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53(2–3):125–157, 1992.

[25] V.Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. The MIT Press, 1984.

Aplicación de la teoría de agentes al modelo de Grafos para la detección de patrones en textos

Fernando Carlos Federico
Facultad de Ingeniería
Universidad de Buenos Aires
fenandofederico1984@yahoo.com.ar

Juan M. Ale
Facultad de Ingeniería
Universidad de Buenos Aires
ale@acm.org

Resumen
Text mining puede ser definido como el descubrimiento de conocimiento en grandes colecciones de documentos. Se asocia principalmente al descubrimiento de patrones interesantes como clusters, asociaciones, desviaciones, similitudes, y diferencias. Por otro lado, los Attributed Relational Graphs (ARG) se definen como una extensión de los grafos ordinarios asociando atributos discretos o reales a sus vértices y arcos. El uso de los atributos permite a los ARG ser posibles de no sólo modelar estructuras topológicas de una entidad sino también sus propiedades no estructurales, que usualmente se pueden representar como vectores. Estas características hacen a esta herramienta un elemento útil a la hora de realizar búsqueda de patrones. Es por ello que, en este trabajo se define un algoritmo basado en grafos para la detección de patrones de textos. Debido a que el volumen de información que se debe procesar es grande, dicho algoritmo contempla la aplicación del modelo de agentes para controlar de manera dinámica el espacio de búsqueda y, en consecuencia, reducir los tiempos de procesamiento de los textos.

Palabras clave: Text Mining, Attributed Relational Graphs, Agentes.

1. Introducción

El motivo de este trabajo se encuentra fundamentado en tres temas pilares, Text Mining, los Attibuted Relational Graphs y el concepto de agentes.

Text mining es un área de investigación emergente que puede ser caracterizada como el descubrimiento de conocimiento en grandes colecciones de documentos, combinando métodos de aprendizaje con métodos de procesamiento de textos. Está asociado principalmente al descubrimiento de patrones interesantes como clusters, asociaciones, desviaciones, similitudes, y diferencias [4, 5,13]. Asimismo, la detección de patrones en textos es una herramienta útil en aplicaciones para reconocimiento inteligente de caracteres, sistemas de compresión de texto, traducciones automáticas, y aplicaciones similares en las que un sistema debe elegir el siguiente elemento (letra, palabra, fonema, etc...) de entre una lista de posibles candidatos [3].

Por otro lado, los Attributed Relational Graphs (ARG) se definen como una extensión de los grafos ordinarios asociando atributos discretos o reales a sus vértices y arcos. El uso de los atributos permite a los ARG ser posibles de no sólo modelar estructuras topológicas de una entidad sino también sus propiedades no estructurales, que usualmente se pueden representar como vectores [17]. Estas características hacen a esta herramienta un elemento útil a la hora de detectar de patrones [14, 15, 16].

Finalmente, existe una gran variedad de definiciones de agentes dependiendo del uso que el autor le haya dado al término. Según la definición de agente de IBM, los agentes son entidades de

software que ejecutan un conjunto de operaciones en nombre de un usuario u otro programa con cierto grado de independencia o autonomía, utilizando algún conocimiento o representación de las metas y deseos del usuario [6]. Sin embargo, investigadores de la Universidad de Indiana forman parte de una comunidad que apoya el enfoque basado en agentes cuyo conocimiento y capacidad deductiva sea limitado. Dicho enfoque propone que un conjunto de agentes simples permite la ejecución de un sistema inteligente de forma más sencilla [2].

En este contexto, el presente trabajo define un modelo genérico basado en los ARG para la búsqueda de patrones, con la incorporación de la teoría de agentes para la reducción de caminos irrelevantes en el grafo. Este comportamiento se modeliza mediante la metodología BDI (Belief, Desire and Intentions) [10], que permite representar el accionar de un agente a partir del conocimiento que posea, las metas y las intenciones del usuario. En resumen, se precisa un algoritmo para detección de patrones que considera algunas características no contempladas en trabajos anteriores (*ver sección 1.2*), a saber:

1. Flexibilidad en el tipo de patrón que se desea detectar.
2. Control del volumen del grafo mediante la reducción de caminos irrelevantes.
3. Velocidad en la detección de patrones.

Se presenta un modelo genérico y fácilmente adaptable al tipo de texto que se desee procesar. Un campo de aplicación posible es el de la clasificación de páginas Web a partir de los patrones de textos que se detecten en ellas (Proyecto de Web Semántica) adaptando el modelo a, por ejemplo, el idioma del sitio y longitud del texto.

El resto del trabajo está organizado de la siguiente manera. En la segunda sección, se define formalmente el grafo que modeliza a un texto para la detección de patrones.. Con el fin de reducir el espacio de búsqueda y agilizar el proceso de detección de patrones, se presenta, en la tercera sección, un método para la selección de los arcos más relevantes del nodo. Para poder llevar adelante dicha tarea mientras se procesa el texto, se consideró a cada uno de los nodos del grafo como un agente cuya meta es la de maximizar el promedio del peso de los arcos que parten de él, dicha modelización se encuentra plasmada en la cuarta sección. La quinta sección presenta un estudio de performance comparando la detección de patrones mediante la utilización de agentes y sin ella. En la sexta sección se presentan las conclusiones y líneas de trabajos futuros.

1.2 Trabajos Relacionados

Existen trabajos en la bibliografía actual que utilizan la representación de textos mediante grafos. Por ejemplo, en (Wei Jin, Rohini Srihari [7]) se propone un método basado en grafos para capturar la estructura y la semántica del documento de forma más efectiva. Básicamente el modelo se basa en un grafo que pondera las relaciones entre los elementos del documento que analiza y extrae conclusiones estructurales a partir de ellas.

En (Tomita,Nakawatase, Ishii [8]), el articulo se concentra principalmente en determinar los algoritmos para obtener subgrafos a partir del grafo original y calcular la similaridad entre ellos. Por otro lado, en (Tomita,Nakawatase, Ishii [3]) se define una arquitectura para representación de paginas Web mediante grafos y su posterior almacenamiento para el descubrimiento de conocimiento en la Web.

Finalmente, existen publicaciones, como (Tsuyoshi Kitani[11]) que basan su trabajo en un idioma en particular (en el caso de [11], el Japonés).

La diferencia de este trabajo con los otros es que se presenta un modelo genérico para la búsqueda de patrones en textos que permite al usuario definir mediante una función peso el tipo de

patrón que desea buscar y que realiza un control sobre el volumen del grafo para obtener mayor velocidad de procesamiento. Es decir, lo determinado en este artículo es independiente del objetivo que tenga el usuario, el idioma o las características del procesamiento que se le quiera dar al texto.

2. Definición del Modelo

2.1 Definición del grafo

Los grafos son estructuras útiles para esquematizar relaciones entre elementos. Es debido a esto que, para la modelización del texto, se utiliza un grafo donde los gramas sean nodos y la relación entre ellos, arcos.

Un grafo es una tripla ordenada $\langle V(G), E(G), \psi_G \rangle$, y la correcta formalización del grafo depende de la definición de la misma. Para ello se exponen las siguientes definiciones.

Definición 1: *Sea* Σ *un alfabeto cualquiera, entonces V (G) es el espacio definido por el producto cartesiano* $V : \Sigma x \Re$.

Acorde a esto, sea un nodo $v \in V$, éste se puede representar como el vector $v = [\alpha, f]$, siendo α un elemento del alfabeto Σ (un grama) y $f \in \Re$ la frecuencia de aparición del grama, dada por la ecuación:

$$f = \frac{N_\alpha}{N_t} \qquad [1]$$

Donde N_α es el número instancias de α (apariciones en el texto), y N_t el número total de instancias en el texto.

Definición 2: *E(G) es el espacio definido por el producto cartesiano* $E : VxVx\Re$.

A partir de esta definición se interpreta que, sea $e \in E, e$ se puede representar como $e = [v_1, v_2, w]$, siendo $v_1, v_2 \in V$ y $w \in \Re$, el peso del arco.

Finalmente, sean $e \in E, u, v \in V$ y $\psi_G(e) = uv$ se dice que e une a u y v, y éstos son extremos de e. En nuestro cado, esta función puede leerse como "*u precede a v en el texto*".

ψ_G cumple las siguientes propiedades:

1) Inyectividad:

$e_1, e_2 \in E$, $u, v \in V, \psi_G(e_1) = \psi_G(e_2) \leftrightarrow e_1 = e_2$

2) Grafo Dirigido:

Sean $e_1, e_2 \in E, u, v \in V, \psi_G(e_1) = uv$ y $\psi_G(e_2) = vu \rightarrow \psi_G(e_1) \neq \psi_G(e_2)$

Ya definida la tripla, solo resta formalizar la definición del grafo que se utilizará a lo largo del documento:

Definición 3 (grafo del modelo): *se define* $H = [V_H, E_H, \psi_G]$ *como un grafo dirigido ponderado, donde* $V_H \subset V$, $E_H \subset E$.

2.2. Función peso ($\xi(d)$)

Los patrones en la red se detectan a partir de los valores de peso de los arcos del grafo. Este valor es modificado según la distancia que se observa entre dos gramas de un texto. La función peso

es la que pondera dicha distancia. No se ha incluido la determinación de una expresión analítica para esta función, debido a que ésta se encuentra sujeta al problema que se desea solucionar.

La función peso también juega un papel preponderante en el número de arcos y nodos de la red, debido a que la ponderación de la distancia es utilizada por el mecanismo de selección para la eliminación de arcos *[sección 3]*.

Debido a que el propósito de esta sección es presentar el objetivo de la función peso, no se define una expresión analítica de la misma. Sin embargo, se expresan las siguientes restricciones

$$\xi(d) > 0 \forall d \in \Re^+ \qquad [2]$$
$$\exists \xi(d) \forall d \in N^+$$

Es ministerio de quién implemente el modelo que se presenta en este capítulo el definir la expresión analítica de $\xi(d)$ acorde a los objetivos que desee cumplir.

2.3. Actualización del grafo

Conforme se recorre un texto para la detección de patrones, el grafo que modeliza dicho documento debe cambiar. La actualización del grafo se puede dar por dos causas:

a) La aparición de un nuevo grama en el texto.
b) La aparición de un grama preexistente.

2.3.1 Aparición de un nuevo grama en el texto

Sea $H^n = (V_H^n, E_H^n)$ un estado n del grafo y $v \in V$ un nuevo vértice, la actualización del grafo se lleva adelante mediante la siguiente función:

$$\Lambda(v) = \bigcup_{v_i \in V_H^n} (v, v_i, t(x_i)) \qquad [3]$$

Siendo t la función definida como $t : \Re^k \rightarrow \Re$

$$t(x_i) = \sum_{j=1}^{k} \xi(x_{ij}) \qquad [4]$$

x_i: vector de distancias entre el grama v y v_i

Debido a que este modelo teórico esta diseñado para una aplicación práctica, se debe hacer mención al valor de k. Dicho número, representa la cantidad de instancias v_i anteriores a v. No obstante, al implementar en un software el algoritmo de actualización, se encuentra inmanejable, en cuestiones de performance y memoria, el almacenamiento de dichas instancias. Es debido a esto que, se introduce al modelo el concepto de *ventana de visualización de gramas* (σ). Dicha *ventana* determina el número de gramas que se deben considerar previos a una actualización del grafo.

Finalmente se define la instancia posterior a la actualización como,

$$H^{n+1} = (V_H^{n+1}, E_H^{n+1}) \qquad [5]$$

Donde,

$$V_H^{n+1} = V_H^n \cup v \qquad \text{y} \qquad E_H^{n+1} = E_H^n \cup \Lambda(v) \qquad [6]$$

2.3.2. Aparición de un grama preexistente (ANE)

Cada vez que aparezca un nodo ya presente en el grafo, se llevará adelante la operación de actualización de pesos que formalmente se define como:

$$\text{Sea } v \in V_H^n, \forall e \in \bigcup_{vi}^{+}, w(e) = w(e) + t(x_{vi}) \quad [1] \; [7]$$

Donde,

x_{vi}: vector de distancias entre dos nodos.

A igual que en la sección anterior, es necesario remarcar que la definición presentada sólo es válida en un marco teórico, debido a que dicha actualización resulta viable computacionalmente sólo si se tiene en cuenta la *ventana de visualización de gramas* que limita el tiempo y memoria necesarios para implementar el algoritmo.

3. Método de selección

3.1. Objetivo

El método de selección que se presenta en este apartado tiene como finalidad reducir el espacio de búsqueda de los patrones de gramas. Este algoritmo permite eliminar aquellos arcos que posean un peso bajo con respecto a la sumatoria total de los pesos del conjunto de arcos salientes del nodo. De esta forma, se eliminan caminos innecesarios y se obtiene un grafo de menor tamaño que facilita el proceso de búsqueda de patrones. Formalmente, el objetivo del algoritmo se puede definir de la siguiente manera:

Definición 4. *Sea* $H = (V, E)$ *un grafo ponderado con función de peso* $w : E \to \Re$ *, encontrar un subgrafo* $H' = (V', E') / \forall v \in V', w(v) > \alpha$ *, donde* α *es una constante del problema.*

3.2. Algoritmo y Estructura de Datos

Se define, en principio, el conjunto de arcos $A = \{a_1, a_2, ..., a_n\}$ y supongamos $A = \bigcup_v^+$. Cada arco de A será evaluado con respecto al conjunto según la función probabilística:

$$\overline{w}^k{}_i = \frac{w^k(a_i)}{\sum_{a_j \in A} w^k(a_j)} \qquad [8]$$

Donde, $w^k(a_j)$ es el peso del arco a_j en una instancia *k* del nodo *v*.

Luego, se especifica $[\overline{w}_{\min}; \overline{w}_{\max}]$ como el rango de valores posibles para $\overline{w}_i$. Es decir, se considera que todo arcos cuyo $\overline{w}$ sea menor a $\overline{w}_{\min}$ no debe ser tomado en cuenta y por ende puede ser eliminado. Mientras que todo arco que tenga un $\overline{w}$ mayor a $\overline{w}_{\max}$ puede ser interpretado como una relación fuerte entre gramas y en consecuencia un patrón.

Cada arco de A es dispuesto en un vector *V* de q posiciones según su $\overline{w}_j$ de la siguiente forma:

[1] Se define como $\bigcup_v^+$ al conjunto de arcos salientes de v.

$$a_j \in V(i) \leftrightarrow \left(\overline{w}_{\min} + (i-1)\left(\frac{\overline{w}_{\max} - \overline{w}_{\min}}{q} \right) \right) < \overline{w}^k{}_j < \left(\overline{w}_{\min} + i.\left(\frac{\overline{w}_{\max} - \overline{w}_{\min}}{q} \right) \right); 1 \le i < q$$

ó

$$a_j \in V(q) \leftrightarrow \left(\overline{w}_{\min} + (q-1)\left(\frac{\overline{w}_{\max} - \overline{w}_{\min}}{q} \right) \right) < \overline{w}^k{}_j \qquad [9]$$

Cada vez que se genera una instancia del nodo v (aparece el grama en el texto) se actualizan todos los pesos de los arcos [*ver sección 3.3.a y 3.3.b*], y en consecuencia los $\overline{w}^k{}_j$. Esto puede generar que, los arcos activados incrementen su posición en el vector. Debido a la presencia de la ventana de visualización de gramas [*ver sección 2.3.a*] cada vez que sucede una instancia del nodo v pueden existir algunos arcos que no se activen y por ende su posición en el vector tiende a disminuir. Es decir, cada vez que aparezca un grama en el texto se debe evaluar a cada uno de los arcos del nodo para verificar su correcta posición en el vector. Debido a que esto es costoso en tiempo, se decidió implementar la siguiente solución: Sucedido un tiempo ε [*ver sección 3.3*] el nivel inferior se limpiará de arcos (se eliminan) y éste pasará a ser el nivel superior del vector.

Por otro lado, cada posición del vector tiene lo que se denomina una zona segura, todo elemento perteneciente a esa zona no descenderá de nivel cuando pase el tiempo ε. La pertenencia de un elemento del vector a la zona de seguridad se explicará en la sección 3.4.

Finalmente, se debe mencionar que el valor de q, como así también, el rango $[\overline{w}_{\min}; \overline{w}_{\max}]$ son parámetros del algoritmo, en consecuencia deben ser definidos por quién lo implemente.

3.3. Definición del tiempo ε

Cada vez que se activa un nodo del grafo (instancia *k*) existe un conjunto de gramas en la ventana de visualización. Cuando se actualizan los pesos de los arcos salientes del nodo sólo se tienen en cuenta dichos gramas. Las distancias existentes entre el nodo y cada uno de ellos determinan un conjunto de incrementos de pesos que denominamos P^k_{vv}.

$$P^k_{vv} = \{p^k_1, p^k_2, \ldots, p^k_\sigma\}$$
$$P^{k+}_{vv} = \sum_{i=1}^{\sigma} p^k_i \qquad [10]$$

Por otro lado, cada nodo *v* del grafo tiene asociado un conjunto de arcos $A = \bigcup_v^+ = \{a_1, a_2, \ldots, a_n\}$. Cada elemento de este conjunto será comparado mediante la expresión *8.* Si tomamos en cuenta dicha expresión y si no hay aumento del peso del arco a_i se observa que:

$$\overline{w}^{k+1}_i = \frac{w^k(a_i)}{\sum_{a_j \in A} w^k(a_j) + P^{(k+1)+}_{vv}} \qquad [11]$$

Si suponemos $P^k_{vv} = P_{vv}$ constante, luego de η instancias se tiene:

$$\overline{w}^{k+\eta}_i = \frac{w^k(a_i)}{\sum_{a_j \in A} w^k(a_j) + \eta * P_{vv}} \qquad [12]$$

Finalmente, otro elemento a tener en cuenta es la posición del arco en el vector V. Como se mencionó en la sección *3.2*, el arco a_i se encuentra en el vector *V* según la forma *9*, para el estudio de ε se debe formalizar los limites de cada posición del vector. Dicho valor es definido de la siguiente manera:

$$\Delta = \Delta(\mathrm{m}) = \left(\frac{\overline{w}_{\max} - \overline{w}_{\min}}{q} \right) \qquad [14]$$

Ya explicitados todos los elementos necesarios, se estudia cuántas instancias del nodo deben ocurrir para que un arco que se encuentre en un nivel determinado del vector disminuya su posición.

Sea τ el valor que debe disminuir el arco a_i para bajar un nivel en el vector y si se supone P_{vv}^k constante se tiene:

$$\tau = \overline{w}_i^k - \overline{w}_i^{k+\eta} \qquad [15]$$

Luego,

$$\eta = \frac{\left(\sum_{a_j \in A} w^k(a_j)\right)\tau}{P_{vv}\left(\overline{w}_i^k - \tau\right)} \qquad [16]$$

Si el arco a_i se encuentra en la posición *m* del vector, entonces τ puede escribirse de la siguiente manera:

$$\varepsilon = \overline{w}_i^k - \theta(\mathrm{m}-1) \qquad [17]$$

Luego,

$$\eta = \frac{\left(\sum_{a_j \in A} w^k(a_j)\right)\ (\%\ w^k - \theta(\mathrm{m}-1))}{P_{vv}\ \theta(\mathrm{m}-1)} \qquad [18]$$

Como norma, se decide esperar lo máximo posible antes de disminuir a un arco de posición. Por los tanto, se elige

$$\theta(m-1) = \overline{w}_{\min}$$
$$\overline{w}_i^k - \theta(m-1) = \Delta$$

Por último se define α_i como la relación entre la sumatoria de los pesos de los arcos eliminados y el total de pesos asociados al nodo.

$$\alpha_i = \frac{\sum_{\Gamma k} w^k(a_i)}{W_j^i} \qquad [19]$$

Donde Γk es el conjunto de pesos de los arcos eliminados en la instancia k.

Así, para la primera iteración:

$$\sum_{a_j \in A} w^1(a_j) = P_{vv}, \text{ y } \alpha_1 = 0 \text{ (no se elimina ningún arco)}$$

Luego,

$$\varepsilon_1 = \frac{\Delta}{w_{\min}}$$

Para la segunda iteración se tiene:

$$\sum_{a_j \in A} w^2(a_j) = \varepsilon_1 P_{vv} - \alpha_1 \varepsilon_1 P_{vv}$$

(Todo lo acumulado en la primera iteración menos lo eliminado)

Por lo tanto,

$$\varepsilon_2 = \varepsilon_1^2 (1 - \alpha_1)$$

Siguiendo el mismo razonamiento, por inducción se tiene:

$$\varepsilon_1 = \frac{\Delta}{w_{min}}$$

$$\varepsilon_2 = \varepsilon_1^{\,2}\,(1-\alpha_1)$$

$$\varepsilon_n = \varepsilon_{n-1}\,(1+(1-\alpha_{n-1})\,\varepsilon_1)$$

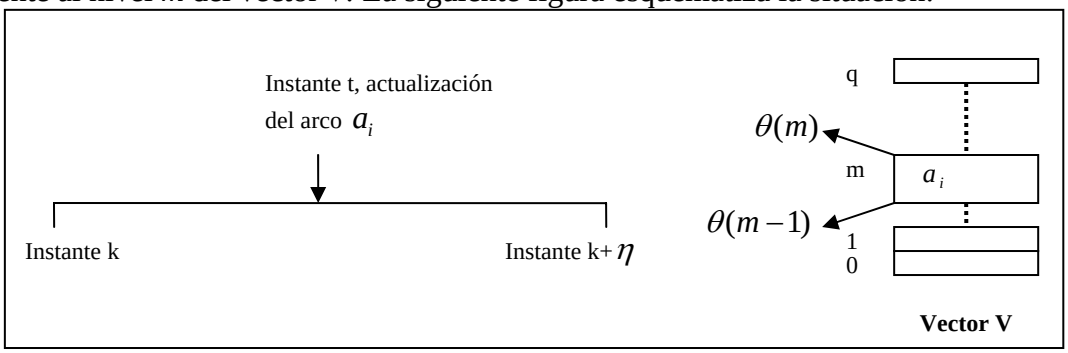
[20]

3.4. Zona de seguridad

Cada posición del vector *V* posee lo que se denomina una zona de seguridad. Todo arco que pertenezca a dicha zona se llama *"arco seguro"*, sin embargo aquel arco que no se encuentre en una zona de seguridad se designa como *"no seguro"*. El manejo de las zonas de seguridad se rige mediante las siguientes reglas:

1. Todo arco que se encuentre en una zona de seguridad no desciende de nivel cuando se eliminan los arcos de la última posición de *V*.
2. Cada vez que se produce una eliminación del nivel inferior todo arco perteneciente una zona de seguridad pierde la condición de *"arco seguro"*, es decir, sale de la zona segura.
3. Siempre que se produzca una eliminación del nivel inferior, toda zona segura se completará con los arcos *"no seguros"* del nivel inmediatamente superior.

En esta sección se define cuánto debe ser el valor del incremento del peso de un arco para entrar en una zona de seguridad. Para ello se sigue el siguiente razonamiento, supongamos que en un instante *k* se definen las η iteraciones que se deben esperar para realizar una nueva eliminación de arcos. Dicho valor se calcula para aquellos arcos que no sean activados en todo el período, es decir, no incrementen su valor peso. Sin embargo, no todos los arcos cumplen con ésta condición, es por ello que suponemos un instante t menor a η donde se produce la activación de un nodo a_i perteneciente al nivel *m* del vector *V*. La siguiente figura esquematiza la situación:



Para que un elemento del vector entre en una zona de seguridad su relación de peso con el total en el instante k+η ($\overline{w}_i^{k+\eta}$) debe ser mayor al límite inferior del nivel donde se encuentra, es decir:

$$\theta(m-1) \le \frac{w^{t-1} + I^t}{W_j^{t-1} + P_{ww} + (\eta - t)\,P_{ww}} \qquad \textit{[21]}$$

Siendo

$$W_j^{\,t-1} = \sum_{a_i \in A} w^{\,t-1}(a_i)$$

$w^{t-1} = w^{t-1}(a_i), a_i \in A$

I^t, el incremento del arco en el instante t

Definimos entonces el valor del incremento para entrar en la zona de seguridad:

$$\theta(m-1) \leq \frac{w^{t-1} + I^t}{W_j^k + t\,P_{vv} + (\eta - t)\,P_{vv}}$$

$$\theta(m-1) \leq \frac{w^{t-1} + I^t}{W_j^k + \eta\,P_{vv}}$$

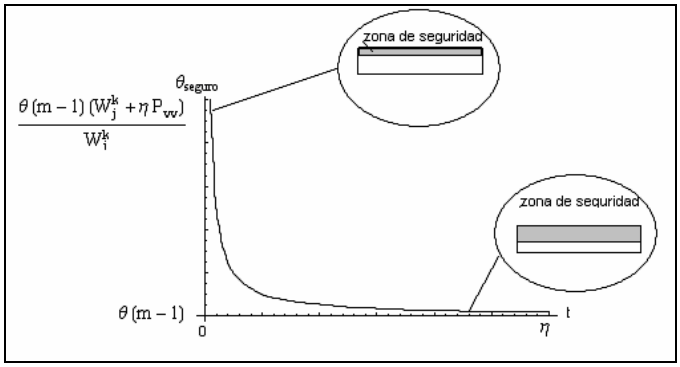
$$\theta(m-1)\,(W_j^k + \eta\,P_{vv}) - w^{t-1} \leq I^t \qquad [22]$$

Es decir, todo incremento en el peso de un arco que cumpla con la condición *22* provocará un pasaje del mismo a la zona de seguridad del nivel. Finalmente encontramos una expresión para la zona de seguridad de un nivel (θ_{seguro}):

$$\frac{w^{t-1} + I^t}{W_j^t} \geq \theta_{seguro} \qquad [23]$$

$$\frac{\theta(m-1)\,(W_j^k + \eta\,P_{vv})}{(W_j^k + t\,P_{vv})} \geq \theta_{seguro}$$

Se puede observar que el tamaño de la zona de seguridad es directamente proporcional al instante t, es decir, acorde suceden instancias del nodo, es más sencillo para un arco entrar a la zona de seguridad.



Se puede ver en la grafica anterior que cada posición del vector posee una zona de seguridad distinta, sin embargo, la variación del tamaño de la misma posee la misma curva independientemente del nivel de *V*.

4. Modelo de Agente

Desde el punto de vista global, el grafo puede entenderse como un conjunto de agentes simples interconectados entre sí. Con el término simples nos referimos a que cada nodo no lleva a cabo un grupo complejo de acciones sino que actúa a partir de primitivas sencillas que permiten, a escala global, desarrollar una actividad compleja. El accionar de cada vértice es independiente de resto por lo que, los nodos, no comparten información sobre el medio que los circunda. La comunicación

interagentes se lleva a cabo sólo una vez terminado el procesamiento del texto. En éste caso, los nodos se comunican entre sí el conjunto de arcos "fuertes" que tienen en posesión.

El funcionamiento interno de los agentes se encuentra explicitado en los párrafos siguientes, donde se especifican las creencias, metas y planes de los mismos.

Creencias: Cada nodo del grafo tiene una visión distinta de su entorno. Es decir, un conjunto diferente de creencias. Es por ello que las variables que aquí se mencionan se encuentran asociadas a un nodo en particular, o lo que es lo mismo, no tienen carácter global en el grafo.

En principio, las variables asociadas al conocimiento del agente pueden diferenciarse *en variables de definición de frecuencias* y *variables de peso*. Las primeras están relacionadas al manejo del tiempo mientras que las segundas se distinguen por su utilidad en la comparación de arcos.

Existen, básicamente, dos variables de frecuencia. Por un lado, la variable λ_v que define el número de instancias del nodo *v* y ε_i que especifica el número de de instancias que se debe aguardar para poder realizar una eliminación de arcos. El cálculo de dicha variable se desarrolló anteriormente en la sección *3.3*.

En lo concerniente a los arcos, cada vértice del grafo tiene conocimiento del conjunto de arcos salientes de él. A partir de este conjunto se pueden definir el segundo tipo de variables, las variables de peso.

Se define como W_v^k a la sumatoria pesos de las instancias de $\bigcup_v^+$:

$$W_v^k = \sum_{a_i \in \bigcup_v^+} w^k(a_i) \qquad [24]$$

Existe, por otro lado, otra variable de peso introducida en la sección *3.2*, particularmente mediante la ecuación *8*. La variable $\overline{w}_v^k$ depende de la instancia del nodo y es la utilizada para la comparación de los arcos.

Es necesario destacar que todo elemento perteneciente a $\bigcup_v^+$ se encuentra en dispuesto en el vector *V* según lo especificado en la sección *3.2*.

Finalmente, cada nodo desconoce absolutamente el estado de su nodo vecino. Sólo finalizado el procesamiento los nodos se comunicarán unos con otros proveyendo a su vecino con los arcos de mayor peso.

Deseos: A diferencia de las creencias, todos los nodos del grafo poseen la misma meta. El objetivo principalmente radica en la maximización del promedio de los pesos que salen de él.

Formalmente,

$$\max\left(\frac{W_v^k}{card\left(\bigcup_v^+\right)}\right)$$

Se puede observar que una buena heurística para el cumplimiento de este objetivo radica en la selección de las mejores instancias de $\bigcup_v^+$. De esta manera se busca eliminar los caminos irrelevantes del grafo y poder, por lo tanto, detectar los patrones del texto de forma más rápida.

Plan: El plan de cada se encuentra regido principalmente por sus variables de frecuencia. El procedimiento básico se resume en la eliminación de arcos irrelevantes según lo determinado en el apartado del proceso de selección *[sección 3]*.

Cada instancia del nodo es determinada por la aparición en el texto del grama que representa. Siempre que esto ocurra se incrementara en uno a la variable λ_v y mientras que ésta sea distinta de ε_i solo se procederá a la actualización de los pesos de los arcos.

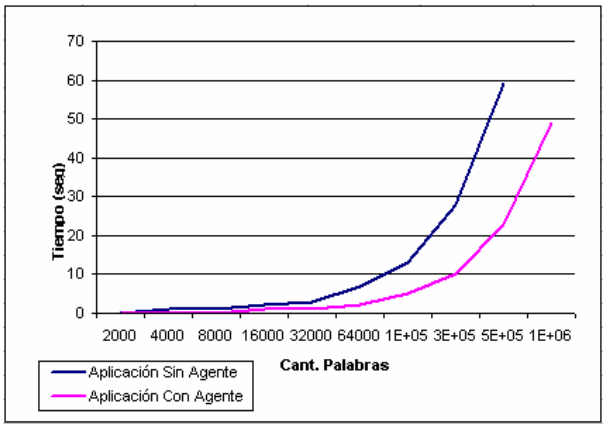
En el caso de que λ_v sea igual a ε_i se procede a eliminar a todos los elementos de la última posición del vector *V*.

Una vez finalizado el documento a procesar, cada nodo del grafo comunicará sus relaciones fuertes al resto para la determinación de los patrones existentes.

5. Caso de Estudio

Para comprobar la efectividad de la aplicación de agentes al grafo se implementó un software basado en lo desarrollado en los capítulos anteriores. Se llevaron adelante dos corridas del mismo, una con la aplicación de agentes y la otra sin ella. En dichas corridas se midieron los tiempos de ejecución y se obtuvieron los resultados graficado en el cuadro. Allí se puede observar que la implementación de agentes al modelo de grafos para la detección de patrones permite una mejora promedio de un 63% en los tiempos.

Cant. Palabras	***Sin Agente***	***Agente***	***Mejora***
2000	0	0	0
4000	1	0	100%
8000	1	0	100%
16000	2	1	50%
32000	3	1	67%
64000	7	2	71%
128000	13	5	62%
256000	28	10	64%
512000	59	23	61%
1024000	-----	49	



Nota: Las pruebas fueron realizadas en un ordenador Pentium 4 2.4 GHz con 512 Mb de memoria. Los textos para realizar las pruebas fueron obtenidos de www.gutenberg.org . Consisten de novelas y cuentos cortos en formato txt idioma ingles.

6. Conclusión y Trabajos Futuros

El método para detección de patrones en textos propuesto permite al usuario definir el tipo de patrón que desea buscar mediante la determinación analítica de la función peso (*sección 2.2*). Esto hace al algoritmo altamente flexible debido a que, mediante la definición de dicha función, se pueden detectar patrones en distintos textos sin importar cuestiones como por ejemplo la longitud o el idioma. Asimismo, la inclusión de la teoría de agentes y el método de selección presentado permite reducir el volumen del grafo disminuyendo los tiempos de procesamiento. Esta cualidad permite que el modelo sea implementado en programas donde se requiera un tiempo de procesamiento corto, como por ejemplo, un explorador de Internet para la clasificación de las páginas Web que muestre.

Entre los trabajos futuros planteados, se encuentran los siguientes:

- Determinación de expresiones analíticas para la función peso dependientes del idioma y el objeto de búsqueda.
- Adaptación del modelo presentado para textos de gran volumen.

- Determinación de métodos de aprendizaje para la eliminación de nodos del grafo poco relevantes.

7. Referencias

[1] Don Adjeroh, Amar Mukherjee, Tim Bell, Matt Powell, Nan Zhang, Proceedings of the Data Compression Conference (DCC '02) , Pattern Matching in BWT-transformed Text, 445, 2002, ISSN:1068-0314

[2] Cristobal Baray y Kyle Wagner Where Do Intelligent Agents Come From?. Crossroad, ACM. 1999.

[3] Christopher D. Manning, Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press: 1999. ISBN 0262133601.

[4] Ciravegna et al., Ed. (2001), Proc. of the 17Th International Joint Conference on Artificial Intelligence (IJCAI-2001), Workshop of Adaptive Text Mining, Seattle, WA, 2001.

[5] Feldman, Ed. (1999), Proc. of The 16th International Joint Conference on Artificial Intelligence (IJCAI-1999), Workshop on Text Mining: Foundations, Techniques and Applications, Stockholm, Sweden, 1999.

[6] Stan Franklin y Art Graesser, “Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents”, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.

[7] Wei Jin, Rohini Srihari, Symposium on Applied Computing archive Proceedings of the 2007 ACM symposium on Applied computing table of contents, Graph-based text representation and knowledge discovery, 807 - 811 ,2007, ISBN: 1-59593-480-4

[8] Junji T., Hidekazu N., Megumi I., Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters WWW Alt. '04,Mayo 2004, ACM Press

[9] Junji T., Hidekazu N., Megumi I., Proceedings of the thirteenth ACM international conference on Information and knowledge management CIKM '04, Noviembre 2004, ACM Press

[10] David Kinny, Michael George, Anand Rao, Proceedings of the 7th European workshop on Modelling autonomous agents in a multi-agent world, A Methodology and Modelling Technique for Systems of BDI Agents, 56-71, 1996, ISBN:3-540-60852-4

[11] Tsuyoshi Kitani, Yoshio Eriguchi, Masami Hara, Journal of Artificial Intelligence Research 2, Pattern Matching and Discourse Processing in Information Extraction from Japanese Text,89-110,1994

[12] M. Marko, M. A. Porter, A. Probst, C. Gershenson, A. Das, 2002, Transforming the World Wide Web into a Complexity-Based Semantic Network, http://arxiv.org/html/cs/0205080 (25/05/2007)

[13] Mladenic, Ed. (2000), Proc. of the Sixth International Conference on Knowledge Discovery and Data Mining, Workshop on Text Mining, Boston, MA, 2000.

[14] -. "Subgraph error correcting isomorphisms for syntactic pattern recognition," IEEE Trans. Sysr., Man, Cybern., vol. SMC-13, no. I , pp. 48-62, Jan. 1983.

[15] W. H. Tsai y K. S. Fu. "Error-correcting isomorphisms of attributed relational graphs for pattern analysis," IEEE Trans. Syst., Man, Cyhern.. vol. SMC-9, pp. 757-768, Dec. 1979.

[16] A. K. C. Wong y S . W. Lu and M. Rioux, "Recognition and shape synthesis of 3-D objects based on attributed hypergraphs," IEEE Trum. Purrern Anal. Machine Intell., vol. PAMI-1 1, no. 3, pp. 279- 289. Sept. 1989.

[17] Dong-Qing Zhang, Shih-Fu Chang. Stochastic Attributed Relational Graph Matching for Image Near-Duplicate Detection, DVMM Technical Report #206-2004-6, Department of Electrical Engineering, Columbia University

Variable Neighborhood Search for solving the DNA Fragment Assembly Problem

Gabriela Minetti
Laboratorio de Investigacin en Sistemas Inteligentes
Universidad Nacional de La Pampa
Argentina
minettig@ing.unlpam.edu.ar

and

Enrique Alba - Gabriel Luque
Dpto. de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Spain
{eat,gabriel}@lcc.uma.es

Abstract

The fragment assembly problem consists in the building of the DNA sequence from several hundreds (or even, thousands) of fragments obtained by biologists in the laboratory. This is an important task in any genome project, since the accuracy of the rest of the phases depends of the result of this stage. In addition, real instances are very large and therefore, the efficiency is also a very important issue in the design of fragment assemblers. In this paper, we propose two Variable Neighborhood Search variants for solving the DNA fragment assembly problem. These algorithms are specifically adapted for the problem being the difference between them the optimization orientation (fitness function). One of them maximizes the Parsons's fitness function (which only considers the overlapping among the fragments) and the other estimates the variation in the number of contigs during a local search movement, in order to minimize the number of contigs. The results show that doesn't exist a direct relation between these functions (even in several cases opposite values are generated) although for the tested instances, both variants allow to find similar and very good results but the second option reduces significatively the consumed-time.

Keywords: DNA Fragment Assembly Problem, Variable Neighborhood Search, 2-opt heuristic

1 INTRODUCTION

DNA fragment assembly is a technique that attempts to reconstruct the original DNA sequence from a large number of fragments. For that, the assembling DNA fragments is divided into three different phases: overlap phase (finding the overlapping fragments), layout phase (finding the order of fragments based on computed similarity scores), and consensus phase (deriving the DNA sequence from the layout). Most of sequence assembly algorithms are based on some variation of the greedy algorithm: Phrap [8], CAP3 [10], Celera assembler [17], TIGR Assembler

[23], STROLL [5]. In the greedy approach, the fragments are assembled by repeatedly merging the pair of fragments with highest overlap according a specific and complex criterion. These methods obtain good results for small-medium sequences but it has some problems in many large genome sequencing projects. Metaheuristic techniques on the contrary are being used with very accurate results even for large problems. Some of them are evolutionary algorithms (EAs) [11, 12, 13, 19], ant colony systems (ACS) [16], simulated annealing (SA) [3, 4, 6, 14, 15], variable neighborhood search (VNS) [9], among others.

Now, we summarize the main features of the most important metaheuristic approaches applied to this problem. Kim and Mohan [11] proposed a parallel version of the Hierarchical Adaptive Genetic Algorithm (HAGA), which adapts its parameters according to the input data, to addresses many specific issues of the DNA fragment assembly. HAGA freezes high confidence subsequences for future iterations of the evolutionary algorithm. Thus the algorithm steps levels, successively reducing the search space size, and working with the blocks identified earlier. Li and Khuri in [12] designed four different techniques to tackle this problem, obtaining the best results with structured pattern matching algorithm (PMA), which is based on the biologist technique called hybridization fingerprinting that deduces the overlap information among DNA clones from biological probes. DNA clones are exact copies of a particular part of a genome and are much longer than fragments. By other side, Alba *et al.* developed different versions to distribute and parallelize metaheuristics and had also created a new heuristic for solving this problem in a very accurate and efficient way. The reader could be obtain more information about those developments in [1, 2, 13, 18].

In [16], Meksangsouy and N. Chaiyaratana presented the use of an ant colony system algorithm in a DNA fragment assembly. They proposed an asymmetric ordering representation where a path co-operatively generated by all ants in the colony represents the search solution. The optimality of the fragment layout obtained is then determined from the sum of overlap scores calculated for each pair of consecutive fragments in the layout.

In the Simulated Annealing approach to the sequence assembly problem, the energy function is based on the overlaps of the fragments. This algorithm tries to minimize this energy function using stochastic reshuffling of the fragments. More details about SA solving FAP can be found in [3, 4, 6].

However, the objective function (fitness function) used in the previous algorithms is very consuming-time. This function evaluates the adaptation degree of found solution to the problem instance maximizing the results. Our objective is to propose a metaheuristic that increases the efficiency without losing quality. For that, we follow the idea presented in [2], we change the optimization orientation estimating the variation of contigs to decrement significatively the consumed-time. The optimal number of contigs for all the instances of this problem is *one*, consequently our new objective is to minimize the number of contigs. In order to do that, we design two versions of the Variable Neighborhood Search, one of them maximizes the overlapping among adjacent fragments in the layout and another minimizes the number of contigs. We show a behavior analysis of both VNS algorithms taking into account the result quality and the consumed CPU time. Finally, we compare our approaches with well-known assemblers in the literature.

The rest of this article is organized as follows. The next section introduces the DNA fragment assembly problem. Section 3 explains how the VNS solves the FAP. Section 4 shows the experiments performed and discusses the results of those experiments. Finally, the last section concludes and provides hints on further research.

2 THE DNA FRAGMENT ASSEMBLY PROBLEM

For studying the functional and structural information of an unknown DNA sequence, biologists compare this sequence with well known ones. If they are similar, they would have the same function. The process of sequencing is called *shotgun sequencing* and was introduced in [21]:

1. The DNA is broken into millions of random fragments.
2. Those fragments are read by a DNA sequencing machine.
3. An assembler pieces together the many overlapping fragments and reconstructs the original sequence.

The last point, the assembling DNA fragments, is divided into three different phases: overlap phase (finding the overlapping fragments), layout phase (finding the order of fragments based on computed similarity scores), and consensus phase (deriving the DNA sequence from the layout). At the assembly stage, the only information available is the sequences of bases, and thus the ordering of the fragments must rely primarily on the similarity of fragments and how they overlap. An important aspect of the general sequencing problem is to determine the relationship and orientation of the fragments. Another important aspect is the incomplete coverage which happens when the algorithm is not able to assemble a given set of fragments into a single contig. A contig is a layout consisting of contiguous overlapping fragments.

Once the fragments have been ordered, the final consensus is generated from the layout. This process includes a detailed alignment step that must considerer the insertion and deletion errors potentially present in the data. To measure the quality of a consensus, we can look at the distribution of the coverage. Coverage at a base position is defined as the number of fragments at that position. It is a measure of the redundancy of the fragment data, and it denotes the number of fragments, on average, in which a given nucleotide in the target DNA is expected to appear. It is computed as the number of bases read from fragments over the length of the target DNA [22]:

$$Coverage = \frac{\sum_{i=0}^{n} length\ \ of\ \ the\ \ fragment\ \ i}{target\ \ sequence\ \ length} \quad (1)$$

where n is the number of fragments. The higher the coverage, the fewer number of the gaps, and the better the result.

Particularly, the assembly of DNA fragments into a consensus sequence corresponding to the parent sequence constitutes the "fragment assembly problem" [22]. It is a permutation and NP-hard problem [20]. Therefore, it is not possible to find an exact algorithm that solves this problem and runs in polynomial time (unless P = NP). In [4, 6], the reader could be find a more detailed information from this process.

3 VARIABLE NEIGHBORHOOD SEARCH FOR SOLVING FAP

The Variable Neighborhood Search (VNS) is a recent metaheuristic presented by Hansen *et al.* in [9]. VNS solves optimization problems by doing systematic changes of neighborhood within a local search. VNS is a descendent method which does not follow a trajectory since explores different predefined neighborhoods of the current solution using a local search (LS). The current solution is changed by a new one if and only if an improvement have been made. The basic idea is to change the neighborhood structure when the local search is trapped on

a local optimum. A neighborhood structure in a solution space S is a mapping $N : S \rightarrow 2^2$, $x \rightarrow N(x)$, where $N(x)$ constitutes the neighborhood of x. The steps of a basic VNS are shown in the Algorithm 1.

Algorithm 1 Basic VNS Algorithm

```
Initialization:
Select the set of neighborhood structures N_k, k = 1, ..., k_max;
Find an initial solution x;
Choose an end condition;
while end condition is not met do
  k = 1;
  while k <= k_max do
    Shaking:
    Generate randomly x' ∈ N_k(x);
    Local Search:
    Obtain the local optimum x'' by applying some local search to x';
    Move or not:
    if x'' is better than x then
      x = x'';
      k = 1;
    else
      k = k + 1;
    end if
  end while
end while
```

Variable Neighborhood Descendent (VND), General VNS (GVNS), and Reduced VNS (RVNS) are extensions of the Basic Variable Neighborhood Search Algorithm. The reader can be find detailed explanations of these extensions in [9, 1].

3.1 Our Proposal

Particularly, we implement two search algorithms based on Basic VNS version according to the permutation representation and the fragment assembly problem. The Algorithm 4 presents this version of VNS.

- *Solution Representation.* We use the permutation representation with integer number encoding. This permutation represents a sequence of fragment numbers, where successive fragments overlap. Consequently each fragment is represented by an unique integer ID. The permutation representation requires special operators to make sure that we always get legal (feasible) solutions. In order to maintain a legal solution, the two conditions that must be satisfied are all fragments must be presented in the ordering, and no duplicate fragments are allowed in the ordering.

- *Initial Solution* The solution used to init this VNS algorithm, is obtained by a method which modifies iteratively the solution applying movements in a structured way as it can see in the Algorithm 2.

- *Neighborhood Structure.* The neighbors in each neighborhood are generated by swapping between two positions from an initial solution. The first position is randomly chosen when

Algorithm 2 Algorithm to create the VNS initial solution

```
for i = 0 to number of fragments - 1 do
  aux_i = random(0, number of fragments * 2);
  x_i = i; {x is the initial solution}
end for
for i = 0 to number of fragments - 2 do
  for j = 0 to number of fragments - 1 do
    if aux_i > aux_j then
      temp = aux_i;
      aux_i = aux_j;
      aux_j = temp;
      temp = x_i;
      x_i = x_j;
      x_j = temp;
    end if
  end for
end for
return x;
```

the neighborhood structure is determined, and it stays fix during the whole execution since it represents to a neighborhood structure. The second position is randomly chosen in the shaking part and it changes in each iteration.

- *Number of neighborhoods*, k_{max}. The number of neighborhoods varies according with the instance. This value is proportional to number of fragments from each instance. In this way, the VNS process creates more search subspaces (neighborhoods) when the number of fragments grows. This feature allows to adequate the VNS computational effort and its efficacy to the problem complexity, intensifying the search when this complexity is increased. This idea arises after many trials where we have tested different ways to establish the number of neighborhoods. We could use the total chromosome size as number of neighborhood but, the execution time in the larger instances grows in a disproportionate way and it is not reflected in the quality of results. We have checked different percentages of instance size (10, 20, 25, 50 and 100%) as number of neighborhoods, and finally we have decided that a 10% is a good compromise value between quality and time.

- *Shaking*. For generating a solution from N_i two positions, of an initial solution copy, are swapped. The first one represents the neighborhood structure and the second one is randomly selected. We use a non consumed-time and simple process (swap) to generate a solution since we only need a solution belonging to a particular neighborhood, which can be modified and improved by a LS.

- *Local Search*. We have worked with a modified version of 2-opt heuristic (see Algorithm 3). We have modified the 2-opt algorithm for reducing the total number of iterations; that results necessary due to the complexity of this method and its application in VNS.

- *Fitness Function*, $F(l)$. Parsons, Forrest, and Burks proposed two different fitness functions which include errors in the sequence information, repeated sequences among other factors [19]. This fitness function sums the overlap score for adjacent fragments in a given solution. When this fitness function is used, the objective is to maximize such score. It

Algorithm 3 Modified 2-opt(x)

```
i = 0;
j = i + 2;
while i < number of fragments do
  x'=x; {x is the initial solution}
  x'_i=x_j;
  x'_j=x_i;
  if x' is better than x then
    x=x';
  end if
  j = j + 1;
  if j > (number of fragments - 1) then
    i = i + 1;
    j = i + 2;
  end if
end while
return initSolution;
```

means that the best individual will have the highest score.

$$F(l) = \sum_{i=0}^{n-2} w(f[i], f[i+1]) \tag{2}$$

where $w_{i,j}$ is the pairwise overlap strength of fragments i and j. The overlap score in F is computed using the semiglobal alignment algorithm.

$F(l)$ is used to evaluate and compare solutions. In this case a solution a is better than another b if $F(l)_a > F(l)_b$. In our first version of VNS, we consider FAP as a maximization problem with $F(l)$ as objective function. This version is called *FVNS*.

- *Contig Estimation*, Δ_c. Alba and Luque proposed to evaluate the candidate solution considering if the number of contigs is incremented or decremented when the local search is applied [2]. In this way only a part of permutation, which is modified by LS, is considered in the evaluation. This evaluation, Δ_c, is very simple since it sums 1 if a contig is broken or rests 1 if two contigs are merged. Therefore, we only need to take care of the concrete positions modified by the variant operator. Considering this, a solution is better than other if its number of contigs is lesser. In other words, FAP is considered as a minimization problem, where minimizing the number of contigs is the objective function. Therefore, we propose the second version of VNS, called *CVNS*, which uses this estimation as fitness function.

- *End condition.* We have established two points in the algorithm where two different stop criterions are necessary. The first point is related with the quantity of times the whole algorithm iterates (first *while* sentence in Algorithm 1); and the second one is related with the stop criterion used in the second *while* sentence in Algorithm 1.

  Since the problem features, we have set the number of complete iterations in only one to reduce the consumed-time. For the second case, we need to limit the number of times that the neighborhoods are explored given that the combination between the neighborhood number (proportional to permutation size) and some instance sizes can produce an

extremely long loop. For that, we have established a maximal iteration number which is calculated as follows:

$$iter_{max} = k_{max} * ((k_{max}/t) + 1) \quad (3)$$

where k_{max} is the neighborhood number and t, is an integer $\in$ [1..10] range.

Algorithm 4 VNS Algorithm for FAP

Initialization:
Set the initial solution x;
Set max as a percentage of the number of fragments;
Select the set of neighborhood structures $N_k, k = 1, ..., k_{max}$;
$k = 0$;
$iter = 0$;
while $((k < k_{max})$ and $(iter < iter_{max}))$ **do**
  Shaking:
  Generate randomly $x' \in N_k(x)$;
  Local Search:
  Obtain the local optimum x'' by applying 2−opt heuristic to x';
  Move or not:
  if x'' is better than x **then**
    $x = x''$;
    $k = 1$;
  else
    $k = k + 1$;
  end if
  $iter = iter + 1$;
end while

4 EXPERIMENTAL RESULTS

In this section we analyze the behavior of our proposed methods. We have chosen three sequences from the NCBI web site[1]: a human MHC class II region DNA with fibronectin type II repeats HUMMHCFIB, with accession number X60189, which is 3,835 bases long; a human apolopoprotein HUMAPOBF, with accession number M15421, which is 10,089 bases long; and the complete genome of bacteriophage lambda, with accession number J02459, which is 20k bases long. We used GenFrag [7] to generate the different data sets shown in Table 1. GenFrag is a UNIX/C application created to accept a DNA sequence as input and to generate a set of overlapping fragments as output, in order to test any assembly application.

Now, we will summarize the results obtained for executing proposed VNS algorithms, which use a 10% of chromosome size as k_{max}. In Table 1 k_{max} and $iter_{max}$ are specified per instance. For each algorithm, we have performed 30 independent runs per instance. We have used a Pentium IV at 2.4 GHz and 1 GB RAM. The operating system used is SuSE Linux with 2.4.19-4GB kernel version. Our aim is to offer meaningful results from a statistical point of view. In Section 4.1, we make an intra comparison on the two algorithms proposed here. After that we compare their results with the obtained ones by another assemblers.

[1] http://www.ncbi.nlm.nih.gov/

Table 1: Information of datasets (Accession numbers are used as instance names) and VNS parameters

Parameters	Instance							
	X60189				M15421			J02459
Coverage	4	5	6	7	5	6	7	7
Fragment Length	395	386	343	387	398	350	383	405
Number of Fragment	39	48	66	68	127	173	177	352
k_{max}	5	6	8	8	14	18	19	36
$iter_{max}$	7	9	13	14	32	52	54	167

4.1 Comparing VNS algorithms between them

First of all, we have considered necessary to analyze the two proposed algorithmic approaches: *FVNS* which optimizes the overlaping among adjacent fragments in the layout and *CVNS* that optimizes the number of contigs. For that, we compare the results from different points of view and perform statistical tests to corroborate our inferences. In the Table 2, we present the best fitness and number of contigs obtained by FVNS and CVNS algorithms in the 30 runs per instance. In Table 3 we show a data resume of FVNS and CVNS for all executions in each instance. They are: the fitness value average, the percentage of the optimal number of contigs, the average of total consumed-time (in seconds) when the optimal number of contigs was found. From each data set, the results of Mann-Whitney U Test are presented, where a '+' symbol indicates the difference between both algorithms is significative; while a '–' symbol indicates that FVNS and CVNS have a similar behavior. Besides we show the percentage of reduced time of CVNS with respect to FVNS.

Table 2: Best fitness and their respective number of contigs, and best number of contigs and their respective fitness obtained by FVNS and CVNS algorithms in all the instances

Instances	*Best (maximal) fitness*		*Best (minimal) number of contigs*	
	FVNS	CVNS	FVNS	CVNS
x60189_4	9920 / 1	2988 / 1	1 / 8629	1 / 1138
x60189_5	12714 / 1	3034 / 1	1 / 12065	1 / 3034
x60189_6	15757 / 2	3343 / 1	1 / 15306	1 / 970
x60189_7	18749 / 1	4892 / 1	1 / 17832	1 / 2305
m15421_5	33720 / 1	3418 / 2	1 / 33495	1 / 2090
m15421_6	40018 / 2	3946 / 2	2 / 38573	2 / 2862
m15421_7	46241 / 2	4724 / 2	2 / 45375	2 / 4724
j02459_7	96816 / 2	5716 / 2	1 / 96517	1 / 5337

Looking at Table 2, we can infer that a good fitness value is not necessarily related with the optimum number of contigs. The reasons for that are the following:

- For some instances, like *x60189* set, *m15421_5* and *j02459_7*, both algorithms find the optimal number of contigs independently if their fitness quality is high or not.
- When FVNS algorithm finds its higher fitness value in *x60189_6* instance, its corresponding number of contigs is greater than one.

- In other instances, as *m15421_5*, CVNS optimizes the number of contigs and their respective fitness values are lesser than the best one. This is clearly visible in Figure 1, where the bars show the fitness value obtained by CVNS in each run and the number on the bar is its corresponding final number of contigs. Particularly, in this figure we can see that some of higher fitness values, are associated with a number of contigs greater than one. By other side, the optimal contigs are related with some of lower fitness values.

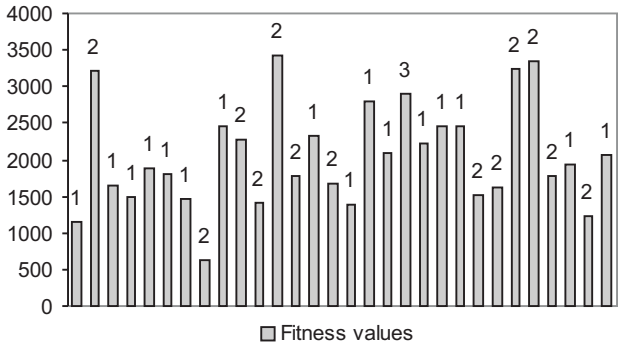


Figure 1: Fitness and number of contigs obtained by CVNS for *m15421_5* instance

Furthermore, from both tables (2 and 3), we can see FVNS outperforms significatively CVNS in all cases when the fitness quality is evaluated. This difference is corroborated using a Mann-Whitney U Test with $\alpha = 0.05$, which is applied to compare the fitness quality obtained by both algorithms in each instance; where the $p-$value is lesser than α. Even although, when we analysis the optimal number of contigs obtained by both algorithms (Table 3), a very similar behavior is observed. For example, for *x60189_4* instance both approaches obtain 1 contig (the optimum) in each execution, but for *x60189_6* instance FVNS reaches the optimum only 93% of times while CVNS finds it every time while for *j02459_7* instance FVNS obtains 1 contig in more executions than CVNS (46.67% against 33% respectively). Those are not significant differences and it is demonstrated by statistical tests where the $p-$values are greater than α. On the other hand, CVNS consumes much lesser time than FVNS to optimize the final number of contigs, in average the percentage of reduced time by CVNS reaches a 98%, being the mean time to obtain the optimal contig 56 seconds. This difference is corroborated using Mann-Whitney U tests with $p-$values lesser than α. The reason for that particularity is directly related with the number of times that fitness function, $F(l)$, is computed by each algorithmic approach. That is, CVNS only calculates this fitness function twice, at the beginning and the end of the algorithm; while the number of evaluations realized by FVNS is equal to iteration number of LS by iteration number of VNS. Thus, in the best cases FVNS surpasses a thousands evaluations of $F(l)$ and this evaluation number depends on the instance size.

Note that we have used a non parametric statistical test, Mann-Whitney U Test, since in many cases we can not prove neither normality or variance homogeneity for using a t-test. The significance level (α) is equal to 0.05 for all tests realized and the probability value ($p-$value) of statistical test is the probability of wrongly rejecting the null hypothesis if it is in fact true. The $p-$value is compared with the significance level and, if it is smaller, the result indicates that compared methods are significatively different.

Table 3: Average of found fitness values, percentage of the optimal number of contigs obtained, average of total time-consumed and time reduction of CVNS with respect to FVNS. The best values are marked in bold.

Instances	*Average Fitness*			*Opt. Contig %*			*Average Total Time*			*% of Reduced time*
	FVNS	CVNS	Mann-Whitney U Test	FVNS	CVNS	Mann-Whitney U Test	FVNS	CVNS	Mann-Whitney U Test	
x60189_4	**9290,47**	1549,80	+	**100,00%**	**100,00%**	–	0,047	**0,002**	+	94.98 %
x60189_5	**11994,07**	1970,73	+	**100,00%**	90,00%	–	0,167	**0,005**	+	96.69 %
x60189_6	**15211,20**	2074,90	+	93,33%	**100,00%**	–	0,846	**0,020**	+	97.65 %
x60189_7	**17966,80**	2770,70	+	96,67%	**100,00%**	–	1,018	**0,022**	+	97.86 %
m15421_5	**32904,00**	2055,20	+	**66,67%**	53,33%	–	46,752	**0,472**	+	98.99 %
m15421_6	**39035,37**	2362,50	+	0,00%	0,00%	–	-	-		
m15421_7	**45596,20**	2832,20	+	0,00%	0,00%	–	-	-		
j02459_7	**95034,50**	3144,80	+	**46,67%**	33,33%	–	11949,845	**55,759**	+	99.53 %
Average	**33379,08**	2345,10		**62,92%**	59,59%		1999,779	**9,380**		97.2 %

4.2 Comparison against other assemblers

In this section we compare the performance of approaches presented here against other assembler algorithms proposed in the literature: Problem Aware Local Search (PALS) [2], a genetic algorithm (GA) [19], a pattern matching algorithms (PMA) [12], and commercially available packages: CAP3 [10] and Phrap [8].

Firstly, we compare CVNS against PALS and we can conclude that these two algorithms present the same behavior for almost instances except for *m15421_6* and *m15421_7* instances.

Now, we compare our approaches with the rest of the above mentioned assemblers, in terms of the final number of contigs assembled (Table 4). In this sense both VNS algorithms have a better or equal behavior than the rest ones. PMA y Phrap obtain the same final number of contigs than our approaches. We can not make an execution time comparison since, in general, the authors does not provide this information.

Table 4: Best final number of contigs for FVNS and CVNS algorithms and for other specialized systems. - symbol indicates that this information is not provided

	FVNS	*CVNS*	*PALS* [2]	*GA* [19]	*PMA* [12]	*CAP3* [10]	*Phrap* [8]
x60189_4	1	1	1	1	1	1	1
x60189_5	1	1	1	1	1	1	1
x60189_6	1	1	1	-	1	1	1
x60189_7	1	1	1	1	1	1	1
m15421_5	1	1	1	6	1	2	1
m15421_6	2	2	-	-	2	2	2
m15421_7	2	2	1	1	2	2	2
j02459_7	1	1	1	13	1	1	1

5 CONCLUSIONS AND FUTURE WORK

In order to design an assembler which finds quickly good solutions for the fragment assembly problem, we propose two search algorithms based on the canonical VNS version. These algorithms are specifically adapted to the problem but they have a different optimization orientation. One of them, *FVNS*, maximizes the Parsons's fitness function which sums the overlap

score for adjacent fragments in a given solution. When this fitness function is used, the objective is to maximize such score. It means that the best individual will have the highest score. Another variant, *CVNS*, evaluates the candidate solution considering if the number of contigs is incremented or decremented when a local search movement is applied. In this way the objective is to minimize the final number of contigs. The result quality of both algorithms is very high since they found the optimal number of contigs in almost instances, whose number of fragments varies in a [39..352] range. Although CVNS reaches these results consuming a 97% less time than FVNS.

Our future work will be to incorporate this new minimization objective in our genetic algorithm versions for the fragment assembly problem.

REFERENCES

[1] E. Alba. *Parallel Metaheuristics A New Class of Algorithms*. WILEY Series on Parallel and Distributed Computing. Wiley, 2005.

[2] E. Alba and G. Luque. A New Local Search Algorithm for the DNA Fragment Assembly Problem. In *Evolutionary Computation in Combinatorial Optimization, EvoCOP'07*, volume 4446 of *Lecture Notes in Computer Science*, pages 1–12, Valencia, Spain, 2007. Springer.

[3] C. Burks, M.L. Engle, S. Forrest, R.J. Parsons, C.A. Soderlund, and P.E. Stolorz. Stochastic Optimization Tools for Genomic Sequence Assembly. In M.D. Adams, C. Fields, and J.C. Venter, editors, *Automated DNA Sequencing and Analysis*, pages 249–259. Academic Press, 1994.

[4] C. Burks, R.J. Parsons, and M.L. Engle. Integration of competing ancillary assertions in genome assembly. In R. Altman, D. Brutlag, P. Karp, R. Lathrop, and D. Searls, editors, *Proceedings Second International Conference on Intelligent Systems for Molecular Biology*, pages 62–69, Menlo Park, CA, 1994. AAA1 Press.

[5] T. Chen and S.S. Skiena. A case study in genome-level fragment assembly. *The Eighth Symposium on Combinatorial Pattern Matching*, pages 206–223, 1997.

[6] G. Churchill, C. Burks, M. Eggert, M.L. Engle, and M.S. Waterman. Assembling DNA Sequence Fragments by Shuffling and Simulated Annealing. Technical Report LA-UR-93-2287, Los Alamos National Laboratory, Los Alamos, NM, 1993.

[7] M.L. Engle and C. Burks. Artificially generated data sets for testing DNA fragment assembly algorithms. *Genomics*, 16, 1996.

[8] P. Green. Phrap. http://www.mbt.washington.edu/phrap.docs/phrap.html, 1996.

[9] P. Hansen, N. Mladenovic, and J.A. Moreno Prez. Variable neighbourhood search. *Revista Iberoamericana de Inteligencia Artificial*, (19):77–92, 2003. ISSN: 1137-3601.

[10] W. Huang and A. Madan. CAP3: A DNA Sequence Assembly Program. *Genome Research*, 9(9):868–877, 1999.

[11] K. Kim and C.K. Mohan. Parallel hierarchical adaptive genetic algorithm for fragment assembly. In IEEE, editor, *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, volume 1, pages 600–607, 2003.

[12] L. Li and S. Khuri. A Comparison of DNA Fragment Assembly Algorithms. In *Proceedings of the 2004 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, pages 329–335, Las Vegas, 2004.

[13] G. Luque, E. Alba Torres, and S. Khuri. *Parallel Algorithms for Bioinformatics*, chapter Chapter 16: Assembling DNA Fragments with a Distributed Genetic Algorithm. Wiley, New York, 2005.

[14] A.P. Lyubartsev, A.A. Martsinovski, and P.N. Vorontosov-Veuaminov. New Approach to Monte Carlo Calculation of the Free Energy: Method of Expanded Ensembles. *Journal of Chemical Physics*, 96:1776–1783, 1992.

[15] E. Marinari and G. Parisi. Simulated Tempering: A new Monte Carlo Scheme. *Europhys. Lett.*, 19:451–458, 1992.

[16] P. Meksangsouy and N. Chaiyaratana. DNA fragment assembly using an ant colony system algorithm. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, volume 3, pages 1756– 1763. IEEE. ISBN: 0-7803-7804-0, 2003.

[17] E. W. Myers. A whole-genome assembly of drosophila. *Science*, 287:219–2204, 2000.

[18] A. J. Nebro, G. Luque, F. Luna, and E. Alba. DNA Fragment Assembly Using a Grid Based Genetic Algorithm. *Computers and Operations Research (to appear)*, 2007.

[19] R. Parsons, S. Forrest, and C. Burks. Genetic Algorithms, Operators, and DNA Fragment Assembly, 1993.

[20] P. Pevzner. *Computational molecular biology: An algorithmic approach*. The MIT Press, 2000.

[21] F. Sanger, A.R. Coulson, G.F. Hong, D.F. Hill, and G.B. Petersen. Nucleotide Sequence of Bacteriophage Lambda DNA. *Journal of Molecular Biology*, 162(4):729–773, 1982.

[22] J. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. International Thomson Publishing, 20 park plaza, Boston, MA02116, 1999.

[23] G. G. Sutton, O. White, M. D. Adams, and A. R. Kerlavage. TIGR Assembler: A new tool for assembling large shotgun sequencing projects. *Genome Science and Technology*, pages 9–19, 1995.

Un Modelo Abstracto de Diálogo Sobre Creencias para Sistemas Multiagente

M. Julieta Marcos Marcelo A. Falappa Guillermo R. Simari
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET),
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial,
Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur,
Avenida Alem 1253,(B8000BCP), Bahía Blanca, Argentina
Tel: (0291) 459-5135 / Fax: (0291) 459-5136
Email: {mjm,mfalappa,grs}@cs.uns.edu.ar

Resumen

Este trabajo muestra una relación entre dos áreas de investigación en Inteligencia Artificial: el modelamiento de *Diálogos en Sistemas Multiagente* por un lado, y la *Teoría de Cambio de Creencias* por el otro. Presentamos un modelo abstracto de diálogo sobre creencias, basado en operadores de cambio no priorizado. La abstracción se refiere tanto al sistema de razonamiento interno utilizado por los agentes, como al tipo particular de diálogo que se quiera modelar.

Básicamente, vemos al diálogo como un proceso mediante el cual los agentes provocan sucesivos cambios sobre una base de conocimiento pública (que representa el *estado* del diálogo). Los agentes tienen *metas* que dictan qué conocimiento exponer en determinado momento.

El modelo impone ciertas restricciones, como por ejemplo que las bases de conocimiento privadas de los agentes no se modifiquen durante el diálogo, y que todos los agentes tengan el mismo grado de credibilidad o autoridad.

Palabras Clave: sistemas multiagente, diálogos entre agentes, cambio de creencias.

1. INTRODUCCIÓN

En un sistema multi-agente los agentes necesitan comunicarse, por diferentes motivos: resolver diferencias de opinión o intereses en conflicto, cooperar para resolver dilemas o encontrar pruebas, o simplemente informarse uno a otro sobre hechos pertinentes. En muchos casos no alcanza con intercambiar mensajes aislados, sino que los agentes necesitan entablar *diálogos* (secuencias de mensajes sobre el mismo tema) [10]. Además, se puede mejorar la calidad de la interacción si los agentes exponen los *argumentos* que justifican lo que dicen [11], es decir, entablan diálogos *basados en argumentación*.

Existe una gran variedad de interacciones con características diferentes que podrían quererse modelar. Una posible tipología, teniendo en cuenta el objetivo común del diálogo y las metas particulares de cada participante, es la siguiente [12]:

Diálogo de Búsqueda de Información. Un agente busca la respuesta a una pregunta en el conocimiento de otro agente. Se supone que este último conoce la respuesta.

Diálogo de Investigación. Todos los agentes colaboran para encontrar la respuesta a una pregunta. Se supone que ninguno de ellos conoce la respuesta.

Diálogo Persuasivo. Un agente trata de convencer a otro para que se adhiera a cierta creencia o punto de vista.

Negociación. Los agentes tratan de llegar a un acuerdo aceptable sobre la división de recursos escasos. Cada uno trata de maximizar su ganancia. La meta del diálogo puede estar en conflicto con las metas individuales de los agentes.

Diálogo Deliberativo. Los agentes colaboran para decidir que acción realizar en cierta situación.

Se puede hacer una distinción entre diálogos *colaborativos* y diálogos *no-colaborativos*. En un diálogo colaborativo los agentes no tienen metas individuales más allá de la meta común del diálogo; por lo tanto, todos colaboran en aras del mismo fin. En un diálogo no-colaborativo, en cambio, los agentes tienen metas individuales que podrían estar en conflicto. Por ejemplo, un diálogo de investigación es colaborativo (el único objetivo compartido por todos los agentes investigadores es descubrir la verdad) pero una negociación no lo es (cada agente negociador tiene como meta maximizar su propia ganancia). Un diálogo deliberativo podría ser colaborativo o no (dependiendo de si los agentes tienen algún interés particular por tomar cierto curso de acción). Un diálogo persuasivo puede ser visto como un diálogo *semi-colaborativo*, donde el agente que persuade tiene una meta particular, pero el persuadido podría no tenerla. Los *agentes colaborativos* expondrán toda la información que consideren relevante, mientras que los *agentes no-colaborativos* podrían ocultar información, sabiendo que es relevante, porque no favorece el cumplimiento de sus metas individuales.

Otra posible diferenciación es entre aquellos diálogos que son *sobre creencias*, y aquellos que no lo son. En un diálogo sobre creencias los participantes hablan sobre la verdad de cierta proposición. A esta categoría corresponden los primeros tres tipos de diálogo: búsqueda de información, investigación y diálogo persuasivo. Sin embargo, un diálogo persuasivo no es *necesariamente* sobre creencias (podría ser, por ejemplo, sobre acciones). Este trabajo está dedicado principalmente a *diálogos sobre creencias*.

Se han realizado varios trabajos con el objetivo de modelar formalmente estas interacciones. Sin embargo, las soluciones propuestas son *ad hoc* y carecen de una fundamentación teórica sólida. En [11], por ejemplo, se investiga un tipo particular de diálogo: la *negociación basada en argumentación*, identificando y describiendo elementos necesarios para su modelamiento (tanto internos como externos a los agentes). En [10], por otro lado, se concentran en *diálogos de investigación*, de *búsqueda de información* y *persuasivos*. Definen un conjunto de *locuciones* para que los agentes puedan intercambiar argumentos, un conjunto de *actitudes* que marcan una relación entre los argumentos que puede construir un agente y las locuciones que puede realizar (intuitivamente, los agentes "menos atrevidos" sólo afirman proposiciones soportadas por "buenos argumentos"), y definen también un conjunto de *protocolos* para llevar a cabo los diálogos. Creemos que una falencia del trabajo citado es que construyen el modelo de diálogo sobre la base de un sistema argumentativo particular.

El objetivo de nuestro trabajo es mostrar como puede construirse un modelo (limitado) de diálogo con un mayor nivel de abstracción, en cuanto al sistema de razonamiento interno utilizado por los agentes. Para ello, utilizaremos el basamento teórico que brinda la *Teoría de Cambio de Creencias*. La misma estudia la *dinámica del conocimiento* en agentes o mundos, es decir, los cambios provocados en una base de conocimiento por el arribo de nueva información. De esta manera, modelaremos el diálogo como un proceso mediante el cual los agentes provocan sucesivos cambios sobre una base de conocimiento pública (que representa el *estado* del diálogo).

El trabajo está estructurado de la siguiente manera. La Sección 2 repasa conceptos básicos de la Teoría de Cambio de Creencias. En la Sección 3 presentamos un modelo parametrizable

de diálogo, basado en operadores de cambio. Se presenta un algoritmo de diálogo, y se analizan algunos aspectos importantes y sus limitaciones. Finalmente, en la Sección 4, se muestra cómo puede utilizarse el algoritmo propuesto para modelar diferentes tipos de interacción.

2. TEORÍA DE CAMBIO DE CREENCIAS

La *Teoría de Cambio de Creencias* estudia la dinámica del conocimiento, esto es, la forma en que se actualiza el conocimiento de un agente después de que recibe información nueva. Un *estado epistémico* es una representación del conocimiento de un agente en un momento del tiempo. Existen, principalmente, dos alternativas para representar estados epistémicos: *conjuntos de conocimiento* [1] o *bases de conocimiento* [6]. Los primeros son conjuntos clausurados bajo algún operador de consecuencia lógica. Los segundos son conjuntos no clausurados, y son los que utilizaremos en este trabajo. Una *actitud epistémica* describe el estado de varios elementos del conocimiento que están contenidos en un estado epistémico. En los modelos clásicos de la Teoría de Cambio se consideran tres actitudes epistémicas: *aceptación*, *rechazo* e *indeterminación*. Una *entrada epistémica* es una pieza de información externa que puede producir *cambios* en un estado epistémico. Las operaciones de *cambio epistémico* que utilizaremos en este trabajo son: *expansión* [1] notada con "+", *mezcla* [4] notada con "$\circ$" y *consolidación* [8] notada con "!". El significado intuitivo de cada una de ellas es el siguiente:

Expansión. Se incorpora conocimiento sin importar si el estado resultante es consistente.

Mezcla. Se combinan dos estados de conocimiento, buscando que el resultado sea consistente.

Consolidación. Se eliminan inconsistencias de un estado de conocimiento.

La *expansión* es la operación más simple. Cuando el estado epistémico se representa con bases, una expansión consiste en una simple unión de conjuntos. Si K es una base de creencias y α una entrada epistémica, entonces la expansión se define como $K + \alpha = K \cup \{\alpha\}$ [6].

La *consolidación* es, en realidad, un caso particular de otra operación: la *contracción* [1]. La operación de contracción elimina una creencia de un estado de conocimiento. En una consolidación la creencia a eliminar es $\perp$ (la contradicción). Entre varios tipos de contracciones, nos enfocaremos en: *Partial Meet Contraction* [1] y *Kernel Contraction* [7]. En base a éstas se definen dos formas de consolidación [8]: *Partial Meet Consolidation* y *Kernel Consolidation*.

Las consolidaciones de tipo *Partial Meet* se basan en *subconjuntos maximales consistentes* y *funciones de selección*. Sea K la base a consolidar. Una *función de selección* selecciona uno o más de todos los subconjuntos maximales consistentes de K. Luego, se define la *consolidación partial meet* como la intersección de todos los subconjuntos elegidos por al función de selección.

Las consolidaciones de tipo *Kernel* se basan en *subconjuntos minimales inconsistentes* y *funciones de incisión*. Sea K la base a consolidar. Una *función de incisión* selecciona una o más creencias para eliminar de cada subconjunto minimal inconsistente de K, buscando restaurar la consistencia del mismo. Luego, se define la *consolidación kernel* como la base original sin las creencias seleccionadas por la función de incisión.

Existen algunas propiedades intuitivas que deberían ser satisfechas por un operador de consolidación [8]:

- *Inclusión.* Para toda base K, debe ser $K! \subseteq K$.
- *Consistencia.* $K!$ debe ser consistente.

- *Relevancia y Retención de Núcleo.* Buscan captar (en menor y mayor medida, respectivamente) la noción de no eliminar de más, es decir, no eliminar creencias que no contribuyen a que la base sea inconsistente.

Luego, pueden definirse las operaciones de consolidación *partial meet* y *kernel* en función de las propiedades anteriores [8]:

- ! es un operador de *partial meet consolidation* si y solo si, para toda base K, ! satisface *inclusión*, *consistencia* y *relevancia*.

- ! es un operador de *kernel consolidation* si y solo si, para toda base K, ! satisface *inclusión*, *consistencia* y *retención de núcleo*.

La propiedad de *relevancia* implica la propiedad de *retención de núcleo*. Por esta razón, todo operador de *partial meet consolidation* es también un operador de *kernel consolidation*. No es cierta la relación inversa.

La operación de *mezcla* puede definirse en términos de la operación de consolidación, y viceversa. Si deseamos combinar una base K con una nueva base H, se define la mezcla como $K \circ H = (K \cup H)!$ [4]. Notemos que no hay inconveniente en hacer una mezcla de bases previamente inconsistentes. También puede definirse la consolidación de K, en términos de una mezcla, como $K! = (K \circ K) = (K \circ \emptyset)$ [4].

La mezcla es un operador de cambio *no priorizado*. Esto significa que no se asigna ninguna prioridad especial al conocimiento nuevo, por lo que podría o no pertenecer al resultado. Otros operadores de cambio no priorizado son, por ejemplo, la *semi-revisión* definida en [8], la *revisión por conjuntos de sentencias* definida en [2], la *revisión selectiva* definida en [3], la *screened revision* definida en [9], etc.. Existe otro tipo de operadores, llamados operadores de cambio *priorizado*, mediante los cuales se asegura que el nuevo conocimiento pertenecerá al estado resultante, como por ejemplo el operador de *revisión* definido en [1]. Estos últimos no se presentarán porque no serán utilizados en este trabajo.

3. UN MODELO DE DIÁLOGO BASADO EN OPERADORES DE CAMBIO

En el modelo que presentaremos, el diálogo se desarrolla en torno a una base de conocimiento pública (nos referimos a ésta como el *estado* del diálogo) que contiene todo el conocimiento expuesto por los agentes hasta el momento. Una *consolidación* del estado del diálogo representa el consenso alcanzado en un momento dado (nos referimos a ésta como el *estado consensuado* del diálogo). Los agentes exponen conocimiento modificando el estado actual, y tienen *metas* que dictan qué conocimiento exponer para lograr el efecto deseado en el estado consensuado del diálogo. Asumiremos los siguientes elementos para construir el modelo:

1. Un *lenguaje de representación de conocimiento* L (al menos proposicional), junto con una *noción de consistencia* de una base de conocimiento $K \subseteq L$, y un *mecanismo de inferencia* ($\vdash$) para derivar conclusiones a partir de subconjuntos de L.

2. *Operadores de cambio* definidos sobre bases de conocimiento $K \subseteq L$: un operador "+" de *expansión*, un operador "!" de *kernel consolidation*, y el operador "$\circ$" de *mezcla* asociado.

3. Un *conjunto de metas* S_G (la noción de *meta* es tratada de manera abstracta).

A continuación introducimos algunas nociones preliminares:

Definición 1 (Agente de Diálogo) Un *agente de diálogo* es un par $A = (K, G)$, donde $K \subseteq L$ es la base de conocimiento privada del agente (asumimos que K es consistente) y $G \in S_G$ es la meta del agente en el diálogo.

Definición 2 (Estado del Diálogo) Un *estado del diálogo* es un conjunto $E \subseteq L$ posiblemente inconsistente.

Definición 3 (Criterio de Éxito) Un *criterio de éxito* es una función booleana φ que toma como entrada un agente de diálogo $A = (G, K)$ y un estado E. Decimos que la meta del agente A *se cumple según* φ en el estado E si $\varphi(A, E) = verdadero$, y que *no se cumple* si $\varphi(A, E) = falso$.

Cuando no haya lugar a dudas sobre el criterio de éxito usado, diremos directamente que la meta del agente A *se cumple* o *no se cumple* en el estado E.

Definición 4 (Entorno de Diálogo) Un *entorno de diálogo* es un par (S_A, φ) donde S_A es un conjunto de agentes de diálogo y φ es un criterio de éxito.

La idea básica es que un agente A_i buscará en su base de conocimiento privada un subconjunto minimal capaz de expandir el estado actual E y obtener un nuevo estado en el que su meta G_i se cumpla. El diálogo termina cuando cada agente o bien alcanza su meta o bien descubre que no tiene medios para alcanzarla (es decir, no existe un subconjunto de su conocimiento capaz de hacer una expansión exitosa). El Algoritmo 1 describe como se desarrollaría un diálogo entre los agentes $A_1 \dots A_n$ con bases de conocimiento $K_1 \dots K_n$ y metas $G_1 \dots G_n$.

Algoritmo 1 : Diálogo entre los Agentes $\{A_1 = (K_1, G_1), \dots, A_n = (K_n, G_n)\}$

1: $E \leftarrow \emptyset$
2: **Repetir**
3: $A_i \leftarrow$ algún agente del conjunto $\{A_1 \dots A_n\}$ tal que G_i no se cumple en E, pero existe $X \subseteq K_i$ (X minimal) tal que G_i se cumple en $E + X$
4: $E \leftarrow E + X$
5: **Hasta** que no exista tal agente A_i
6: **Retornar** E!

Podemos ver que hay elementos sin especificar en este algoritmo. La noción de *meta* y el *criterio de éxito* de una meta en un estado son tratados en forma abstracta porque dependen del tipo de diálogo que se quiera modelar. En la sección siguiente mostraremos una forma de definir estos elementos para modelar algunos tipos de diálogos.

El operador de consolidación adecuado también depende de las características particulares de la interacción que se quiera modelar, por eso su implementación no es especificada en el modelo. El modelo sí especifica que el cambio es *no priorizado*. Sin embargo, existen algunas situaciones en las que sería más adecuado un operador de cambio *priorizado*. Supongamos que un agente A tienen más autoridad o credibilidad que otro agente B, entonces el conocimiento que expone A debería tener prioridad con respecto al conocimiento que expone B. Por otro lado, supongamos que un agente rectifica su conocimiento durante el diálogo porque recibió percepciones (provenientes de una fuente externa al diálogo) más acertadas , entonces las nuevas

creencias expuestas por el agente deberían tener más prioridad que las creencias previas expuestas por ese mismo agente. Por lo tanto, el uso de cambio no priorizado en este trabajo se justifica en las siguientes suposiciones:

1. Todos los agentes tienen el mismo grado de autoridad (o credibilidad).

2. Los agentes no reciben nuevas percepciones (externas al diálogo) mientras dialogan. Es decir, sus bases de conocimiento internas no se modifican durante el diálogo.

Otro aspecto observable en este algoritmo es que es *no determinístico*, por dos razones: pueden existir varios agentes A_i, y varios conjuntos $X \subseteq K_i$, que verifican la condición del paso 3. Diferentes caminos pueden conducir a resultados distintos. Si bien los agentes no se turnan estrictamente para hablar, se puede asegurar que un agente nunca hablará dos veces consecutivas (ya que su meta se cumple inmediatamente después de hablar).

La ejecución termina cuando se alcanza un estado en el que cada agente, o bien cumple su meta o bien descubre que no puede cumplirla.

Definición 5 (Estado Final) Un estado E es un *estado final* para un entorno de diálogo (S_A, φ) si y solo si para cada agente $A_i = (K_i, G_i) \in S_A$ se verifica alguna de las condiciones siguientes:

1. G_i se cumple en E, o bien

2. No existe $X \subseteq K_i$ tal que G_i se cumpla en $E + X$

Es fácil ver que el algoritmo siempre termina. En un caso extremo, los agentes exponen la totalidad de su conocimiento, alcanzando un estado final. El resultado devuelto por el algoritmo es la consolidación del último estado alcanzado, y esto debe interpretarse como el consenso al que llegaron los agentes mediante el diálogo. La siguiente definición relaciona un entorno de diálogo con un posible resultado devuelto por el algoritmo.

Definición 6 (Diálogo) Un *diálogo* es un par (Γ, E) donde Γ es un entorno de diálogo y E es un estado final para Γ.

El estado del diálogo mantiene *todo* el conocimiento que ha sido expuesto (notemos que en el paso 4 del algoritmo el estado E se modifica mediante una *expansión*). Hacer *expansiones* en lugar de *mezclas* brinda dos ventajas: (1) los agentes pueden hacer uso implícito del conocimiento publicado por otros con anterioridad, y (2) se controla implícitamente que los agentes no cometan *falacias* [5] en el diálogo.

Con respecto al estado inicial del diálogo (llamémoslo E_0), si bien en el algoritmo anterior asumimos $E_0 = \emptyset$, esto podría ser de otra manera si resultara más adecuado. Supongamos, por ejemplo, que los agentes recuerdan diálogos pasados. Entonces podrían comenzar el diálogo con un conjunto no vacío de conocimiento público, proveniente de diálogos anteriores entre esos mismos agentes.

Finalmente, podemos destacar algunas restricciones impuestas por el modelo:

1. El *conocimiento público* es un subconjunto del *conocimiento privado* de los agentes. Es decir, que los agentes no dicen nada que no forme parte explícitamente de sus bases de conocimiento privadas.

2. Los agentes no pueden retractarse arbitrariamente sobre locuciones pasadas, sino sólo a través del operador *no priorizado* de mezcla.

3. No se modifican las bases de conocimiento privadas de los agentes. El conocimiento que expone un agente sólo se ve reflejado en el estado público del diálogo, pero no afecta a las bases privadas de otros agentes.

Creemos que estas restricciones no impiden modelar con naturalidad la mayoría de los diálogos sobre creencias, pero podrían ser demasiado fuertes para otros tipos de diálogo, como la negociación o el diálogo deliberativo.

4. MODELANDO DIFERENTES TIPOS DE DIÁLOGO

El Algoritmo 1 intenta capturar el esquema general de cualquier tipo de diálogo: en un momento dado (un estado del diálogo) un participante considera que debe exponer cierto conocimiento, entonces lo hace y se produce un cambio de estado. Creemos que la diferencia entre distintos tipos de diálogo radica, en parte, en el criterio usado por los participantes para determinar si tienen algo para decir (y para elegir qué decir); y usamos la noción de *meta* para representar ese criterio. Veremos como pueden modelarse algunos tipos de diálogo de los presentados en la Sección 1 definiendo de manera adecuada las metas de los agentes. Por ejemplo, en un diálogo persuasivo la meta del agente que persuade podría ser una sentencia $\alpha \in L$ y el criterio de cumplimiento, para un estado E, podría ser $E! \vdash \alpha$. En este trabajo usaremos la siguiente representación de metas:

Definición 7 (Meta) Una *meta* es un par $(\alpha, actitud)$, con $\alpha \in L$ y $actitud \in \{\oplus, \ominus, ?\}$, donde $\oplus$ representa una actitud *a favor* de α, $\ominus$ representa una actitud *en contra* de α, y ? representa una actitud *imparcial* con respecto a α.

De esta manera queda definido un conjunto S_G de metas para un lenguaje L. En lo que resta del trabajo, asumiremos siempre este conjunto de metas.

No deben confundirse estas actitudes de los agentes en el diálogo con las *actitudes epistémicas* mencionadas en la Sección 2. Si bien existe cierta relación entre las actitudes epistémicas de aceptación/rechazo y las metas $(\alpha, \oplus)/(\alpha, \ominus)$, no sucede lo mismo con la actitud epistémica de indeterminación y la meta $(\alpha, ?)$. La siguiente definición clarifica el significado de las actitudes mencionadas en la Definición 7.

Definición 8 (Primer Criterio de Exito) Definimos el siguiente criterio φ_1 de cumplimiento de metas:

- una meta $G{=}(\alpha, \oplus)$ se cumple en un estado $E \subseteq L$ si y solo si $E! \vdash \alpha$
- una meta $G{=}(\alpha, \ominus)$ se cumple en un estado $E \subseteq L$ si y solo si $E! \vdash \neg\alpha$
- una meta $G{=}(\alpha, ?)$ de un agente $A = (K, G)$ se cumple en un estado $E \subseteq L$ si y solo si se verifican las siguientes condiciones:
  1. $E! \vdash \alpha \Leftrightarrow (E \circ K) \vdash \alpha$
  2. $E! \vdash \neg\alpha \Leftrightarrow (E \circ K) \vdash \neg\alpha$

Observación 1 Dada una sentencia α, una meta $(\alpha, ?)$ es equivalente a una meta $(\neg\alpha, ?)$, y una meta $(\alpha, \ominus)$ es equivalente a una meta $(\neg\alpha, \oplus)$.

Intuitivamente, si un agente tiene una meta $(\alpha, \oplus)$ significa que tiene una inclinación particular por concluir que la sentencia α es verdadera. Por el contrario, si un agente tiene una meta $(\alpha, \ominus)$ significa que tiene una inclinación particular por concluir que la sentencia α es falsa. Por último, si un agente tiene una meta $(\alpha, ?)$ significa que su actitud en el diálogo es imparcial y su único objetivo es averiguar la verdad sobre la sentencia α. Las metas $(\alpha, \oplus)$ y $(\alpha, \ominus)$ son adecuadas para agentes no-colaborativos, mientras que las metas $(\alpha, ?)$ son adecuadas para agentes colaborativos.

Nos concentraremos en diálogos en los cuales todos los agentes tienen metas referidas a la misma sentencia α, como se define a continuación.

Definición 9 (Entorno de Diálogo Sobre α) Sea $\alpha \in L$. (S_A, φ) es un *entorno de diálogo sobre* α si y solo si $G_i = (\alpha, actitud_i)$ para todo agente $A_i \in S_A$.

Definición 10 (Diálogo Sobre α) (Γ, E) es un *diálogo sobre* α si y solo si Γ es un entorno de diálogo sobre α.

Las metas $(\alpha, ?)$ apuntan (aunque no siempre lo logran, como veremos más adelante) a obtener diálogos con una propiedad especial: que las conclusiones resultantes coincidan con lo que se concluiría de la unión consolidada de las bases de conocimiento privadas de los agentes. Formalizamos esta propiedad de los diálogos con la siguiente definición.

Definición 11 (Diálogo Completo) Un diálogo sobre α (Γ, E), con $\Gamma = (\{A_1 = (K_1, G_1), \ldots, A_n = (K_n, G_n)\}, \varphi)$, es un *diálogo completo* si y solo si se verifican las siguientes condiciones:

1. $E! \vdash \alpha \Leftrightarrow (\bigcup_{1 \leq i \leq n} \{K_i\})! \vdash \alpha$
2. $E! \vdash \neg\alpha \Leftrightarrow (\bigcup_{1 \leq i \leq n} \{K_i\})! \vdash \neg\alpha$

Ahora podemos definir formalmente algunos tipos de diálogo:

Definición 12 (Diálogo de Investigación) Sea $\alpha \in L$. Un *Diálogo de Investigación* (sobre α) es un diálogo sobre α en el cual todos los agentes participantes tienen la misma meta $(\alpha, ?)$.

Definición 13 (Diálogo de Búsqueda de Información) Un *Diálogo de Búsqueda de Información* (sobre α) es un Diálogo de Investigación (sobre α) en el cual existe por lo menos un agente participante $A_i = (K_i, G_i)$ tal que: o bien $K_i \vdash \alpha$ o bien $K_i \vdash \neg\alpha$.

Definición 14 (Diálogo Persuasivo) Sea $\alpha \in L$. Un *Diálogo Persuasivo* (sobre α) es un diálogo sobre α en el cual existe por lo menos un agente participante $A_i = (K_i, G_i)$ tal que $G_i = (\alpha, actitud_i)$ y $actitud_i \neq ?$.

Esta definición no respeta exactamente la caracterización previa (Sección 1) de Diálogo de Investigación, en la cual se menciona que ningún participante conoce la respuesta a la pregunta en cuestión. Decidimos adoptar esta visión más general y ver al Diálogo de Búsqueda de Información como un caso particular de Investigación.

Las metas y diálogos así definidos son solamente ejemplificaciones sobre como podrían modelarse algunas interacciones con el modelo propuesto. A continuación, se ilustra con ejemplos de Diálogos de Investigación y Diálogos Persuasivos el funcionamiento del Algoritmo 1, y se muestran algunos problemas que podrían surgir. Por simplicidad consideramos diálogos entre dos agentes y utilizamos *Lógica Clásica Proposicional* como lenguaje de representación de

conocimiento. Las consolidaciones son realizadas de manera arbitraria, dado que esta simplificación no afecta la ilustratividad de los ejemplos. La representación de metas es de acuerdo a las Definiciones 7 y 8.

Ejemplo 1 Supongamos un *Diálogo de Investigación* en un entorno ($\{A_1 = (K_1, G_1)$ y $A_2 = (K_2, G_2)\}$, φ_1) con:

$K_1 = \{a, a \rightarrow c, b \rightarrow c, c \rightarrow d\}$ $\quad$ $K_2 = \{a \wedge c \rightarrow \neg d, b \wedge c \rightarrow d\}$ $\quad$ $G_1 = G_2 = (d, ?)$

Supongamos además que el operador ! de *partial meet consolidation* se comporta de la manera especificada a continuación, cuando es aplicado al siguiente estado:

$\{a, a \rightarrow c, c \rightarrow d\ , a \wedge c \rightarrow \neg d\}! \vdash \neg d$ $\quad$ *(1)*

De acuerdo a esto, se muestra a continuación una posible ejecución del algoritmo. En cada paso indicamos qué agente interviene y qué conocimiento publica (respetando una posible elección del paso 3 del Algoritmo 1), y también indicamos el efecto en el estado consensuado del diálogo. Notaremos con E_i el estado del diálogo en la iteración i:

1. $E_1 = \emptyset$
2. El agente A_1 dice: $\{a, a \rightarrow c, c \rightarrow d\}$
   $E_2! \vdash d$
3. El agente A_2 dice: $\{a \wedge c \rightarrow \neg d\}$
   $E_3! \vdash \neg d$
4. Termina el diálogo en el estado E_3

El diálogo termina en el estado E_3 porque ambos agentes cumplen sus metas. En general, podemos decir que:

Observación 2 En un diálogo de investigación todos los agentes cumplen sus metas al terminar el diálogo (es fácil ver que si una meta $G_i = (\alpha, ?)$ de un agente A_i no se cumple en E entonces debe existir $X \subseteq K_i$ tal que G_i se cumple en $E + X$).

En este caso se concluye, por *(1)*, que la sentencia d es falsa. Esta misma conclusión se obtiene de $(K_1 \cup K_2)!$ (el diálogo es completo). El Ejemplo 2 muestra que esto no siempre es así. Hay casos en los que los agentes quedan en una situación de *bloqueo* sin poder exponer toda la información relevante (porque no advierten que es relevante).

El Ejemplo 3 muestra un diálogo de investigación en el que, a diferencia de este, la conclusión final contradice la opinión individual de todos los agentes.

Ejemplo 2 Supongamos que modificamos el entorno del Ejemplo 1 agregando sentencias a ambas bases:

$K_1 = \{a, a \rightarrow c, b \rightarrow c, c \rightarrow d, f \wedge g \rightarrow \neg d\}$ $\quad$ $K_2 = \{a \wedge c \rightarrow \neg d, b \wedge c \rightarrow d, f, f \rightarrow g, g \rightarrow d\}$
$G_1 = G_2 = (d, ?)$

Supongamos además que el operador ! de *partial meet consolidation* se comporta de la manera especificada a continuación, cuando es aplicado a cada uno de los estados siguientes:

$\{a, a \rightarrow c, c \rightarrow d, a \wedge c \rightarrow \neg d\}! \vdash \neg d$ *(1)*
$\{a, a \rightarrow c, c \rightarrow d, a \wedge c \rightarrow \neg d, f, f \rightarrow g, g \rightarrow d\}! \vdash d$ *(2)*
$\{a, a \rightarrow c, c \rightarrow d, a \wedge c \rightarrow \neg d, f, f \rightarrow g, g \rightarrow d, f \wedge g \rightarrow \neg d\}! \vdash \neg d$ *(3)*

De acuerdo a esto, se muestra a continuación una posible ejecución del algoritmo.

1. $E_1 = \emptyset$
2. El agente A_1 dice: $\{a, a \rightarrow c, c \rightarrow d\}$
   $E_2! \vdash d$
3. Termina el diálogo en el estado E_2

En este caso podemos ver que el diálogo resulta en la aceptación de la sentencia d, pero si ambos agentes expusieran todo su conocimiento se obtendría una conclusión diferente. Notemos que el agente A_2 considera, en el estado E_2, que no tiene nada relevante para decir. Esto se debe a que dicho agente advierte, por *(2)*, que aún aportando todo su conocimiento no cambiaría la conclusión sobre d. Lo que el agente A_2 no alcanza a advertir es que, por *(3)*, la conclusión *sí* cambiaría en una iteración posterior, luego de la intervención del agente A_1. Esta situación (la llamamos situación de *bloqueo*) es poco deseable para agentes con metas $(\alpha, ?)$, ya que estos agentes buscan idealmente diálogos completos. Observemos que, por *(1)*, existe en este caso un subconjunto propio del conocimiento privado del agente A_2 capaz de cambiar la conclusión pública sobre d. Una forma de reducir las situaciones de bloqueo es redefinir el criterio de cumplimiento de las metas $(\alpha, ?)$ de la siguiente manera:

Definición 15 (Redefinición del Criterio de Éxito de Metas $(\alpha, ?)$**)** Una meta $G{=}(\alpha, ?)$ de un agente $A = (K, G)$ se cumple en un estado $E \subseteq L$ si y solo si, para todo $X \subseteq K$, se verifican las siguientes condiciones:

1. $E! \vdash \alpha \Leftrightarrow (E \circ X) \vdash \alpha$
2. $E! \vdash \neg\alpha \Leftrightarrow (E \circ X) \vdash \neg\alpha$

Llamaremos φ_2 el criterio de éxito φ_1 modificado según la Definición 15. El cumplimiento de una meta $(\alpha, ?)$ según φ_2 implica trivialmente el cumplimiento de una meta $(\alpha, ?)$ según φ_1. La diferencia entre un diálogo de investigación con el criterio φ_1 y uno con el criterio φ_2 es que en el segundo caso los agentes publicarán más información, evitando algunas situaciones de bloqueo.

Ejemplo 3 Reconsideremos el Ejemplo 2, pero ahora usando el criterio φ_2 en lugar de φ_1:

$K_1 = \{a, a \rightarrow c, b \rightarrow c, c \rightarrow d, f \wedge g \rightarrow \neg d\}$ $\quad$ $K_2 = \{a \wedge c \rightarrow \neg d, b \wedge c \rightarrow d, f, f \rightarrow g, g \rightarrow d\}$
$G_1 = G_2 = (d, ?)$

A continuación se muestra una posible ejecución del algoritmo, asumiendo el mismo operador de consolidación que en el Ejemplo 2:

1. $E_1 = \emptyset$
2. El agente A_1 dice: $\{a, a \rightarrow c, c \rightarrow d\}$
   $E_2! \vdash d$

3. El agente A_2 dice: $\{a \wedge c \rightarrow \neg d, f, f \rightarrow g, g \rightarrow d\}$
   $E_3! \vdash d$

4. El agente A_1 dice: $\{f \wedge g \rightarrow \neg d\}$
   $E_4! \vdash \neg d$

5. Termina el diálogo en el estado E_4

En este caso el diálogo termina concluyendo que la sentencia d es falsa, a pesar de que cada agente creía individualmente lo contrario. El resultado coincide con lo que se concluiría de $(K_1 \cup K_2)!$ (el diálogo es completo).

Veremos con el Ejemplo 4 que en algunos casos la situación de bloqueo persiste a pesar de la redefinición de criterio del éxito.

Ejemplo 4 Supongamos un caso de *Diálogo de Investigación* extremadamente simple:

$K_1 = \{a\} \qquad K_2 = \{a \rightarrow b\} \qquad G_1 = G_2 = (b, ?)$

Sería deseable concluir que la sentencia b es verdadera. Sin embargo, el modelo de diálogo propuesto no logra obtener esta conclusión (ya sea con el criterio de éxito φ_1 o φ_2). Se puede ver fácilmente que se genera un diálogo vacío en el que ninguno de los agentes puede empezar a hablar.

Ejemplo 5 Por último, veremos un ejemplo de *Diálogo Persuasivo*. Tomaremos el entorno del Ejemplo 3 pero modificaremos la meta del agente A_1, de la siguiente manera:

$K_1 = \{a, a \rightarrow c, b \rightarrow c, c \rightarrow d, f \wedge g \rightarrow \neg d\} \qquad K_2 = \{a \wedge c \rightarrow \neg d, b \wedge c \rightarrow d, f, f \rightarrow g, g \rightarrow d\}$
$G_1 = (d, \oplus) \quad G_2 = (d, ?)$

A continuación se muestra una posible ejecución del algoritmo:

1. $E_1 = \emptyset$

2. El agente A_1 dice: $\{a, a \rightarrow c, c \rightarrow d\}$
   $E_2! \vdash d$

3. El agente A_2 dice: $\{a \wedge c \rightarrow \neg d, f, f \rightarrow g, g \rightarrow d\}$
   $E_3! \vdash d$

4. Termina el diálogo en el estado E_3

La conclusión alcanzada es que la sentencia d es verdadera. Podemos ver que el diálogo no es completo. Sin embargo, la incompletitud no es un problema en este caso, sino que es causada intencionalmente por el agente A_1 para cumplir su meta individual.

5. CONCLUSIONES Y TRABAJO FUTURO

El problema abordado en este artículo es el modelamiento de diálogos en sistemas multiagente. Propusimos un algoritmo abstracto no determinístico que simula una interacción entre dos o más participantes. Creemos que los siguientes son *aspectos positivos* de la solución propuesta: (1) el uso de *Operadores de Cambio* nos permite abstraernos de la teoría lógica subyacente (lenguaje y mecanismo de inferencia), y (2) el uso de la noción abstracta de *meta* nos permite

parametrizar las actitudes de los agentes en el diálogo, abstrayéndonos, en cierta medida, del *tipo* de diálogo.

Identificamos también algunas *limitaciones* del modelo propuesto: (1) está principalmente orientado a *diálogos sobre creencias*, (2) todos los agentes tienen el mismo grado de autoridad (o credibilidad), (3) los agentes no reciben nuevas percepciones (externas al diálogo) mientras dialogan, y (4) el trabajo brinda un aporte teórico, pero no práctico, dada la complejidad computacional del algoritmo propuesto.

Nuestro trabajo futuro estará orientado a: (1) investigar el modelamiento de otros tipos de diálogo (como la negociación y el diálogo deliberativo), (2) analizar el uso de *lógicas argumentativas* como lenguaje de representación de conocimiento (creemos que esto podría facilitar el hallazgo del conjunto X del paso 3 del Algoritmo 1), (3) buscar estrategias para solucionar las situaciones de bloqueo mencionadas en la Sección 4, y (4) profundizar sobre implementaciones adecuadas del operador de consolidación en diferentes tipos de diálogo.

REFERENCIAS

[1] C. Alchourrón, P. Gärdenfors, and D. Makinson. *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. The Journal of Symbolic Logic, 50:510-530, 1985.*

[2] M. A. Falappa, G. Kern-Isberner, and G. R. Simari. *Belief Revision, Explanations and Defeasible Reasoning. Artificial Intelligence Journal, 141:1-28, 2002.*

[3] E. L. Fermé and S. O. Hansson. *Selective Revision. Studia Logica, 63:331-342, 1998.*

[4] A. Fuhrmann. *An Essay on Contraction. Studies in Logic, Language and Information, CSLI Publications, Stanford, California, 1997.*

[5] C. L. Hamblin. *Fallacies. Methuen and Co Ltd, London, 1970.*

[6] S. O. Hansson. *Belief Base Dynamics. PhD thesis, Uppsala University, Department of Philosophy, Uppsala, Sweden, 1991.*

[7] S. O. Hansson. *Kernel Contraction. The Journal of Symbolic Logic, 59:845-859, 1994.*

[8] S. O. Hansson. *Semi-Revision. Journal of Applied Non-Classical Logic, 7:151-175, 1997.*

[9] D. Makinson. *Screened Revision. Theoria: Special Issue on Non-Prioritized Belief Revision, 1997.*

[10] S. Parsons , M. Wooldridge , and L. Amgoud. *Properties and Complexity of Some Formal Inter-agent Dialogues. Journal of Logic and Computation, 13:347-376, 2003.*

[11] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. *Argumentation-based negotiation. The Knowledge Engineering Review, 18:343-375, 2003.*

[12] D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany, NY, 1995.*

Desarrollo de Sistemas Inteligentes aplicados a redes eléctricas industriales

Andrés Krapf y Ana Casali
{akrapf,acasali}@fceia.unr.edu.ar

Depto. de Sistemas e Informática
Facultad de Cs.Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario
Av. Pellegrini 250. 2000 Rosario, Argentina.

Abstract

Este trabajo sintetiza el desarrollo de un Sistema Inteligente de Control (SIC) para la Automatización de Sistemas Eléctricos de Potencia (SEPs) del tipo industrial. Las tareas automatizadas son detección y aislación de fallas, y automatización de métodos de back-up que soporten posibles pérdidas de integridad en los componentes involucrados. Para este sistema se propone una arquitectura multiagente altamente escalable y flexible, y se decriben los agentes necesarios para realizar las distintas tareas. Se implementan prototipos de dos de los agentes que integran el SIC utilizando el modelo BDI (Belief-Desire-Intention) y se evalúa su comportamiento.

Keywords: Diseño Orientado a Agentes, Agentes BDI, Sistemas Eléctricos de Protencia.

1. INTRODUCCIÓN

El objetivo de un Sistema Eléctrico de Potencia (SEP) es proveer energía eléctrica con criterios de calidad, seguridad y fiabilidad acorde a las normas vigentes.

En este trabajo se hace especial referencia a las redes eléctricas industriales[1]. En la actualidad estos sistemas presentan la posibilidad de mejorar la calidad de prestación. Para ello es necesario disponer de medios de protección y control que se adapten a distintas condiciones de hardware y de operación del SEP. La condición básica de todo SEP es el equilibrio entre generación y demanda. El mundo y en especial la Argentina se enfrenta al problema que los recursos energéticos son limitados y el margen de reserva de generación es cada vez menor, por lo que es necesario agudizar los mecanismos de control para evitar colapsos generales o parciales.

Para los ingenieros de SEPs el principal factor a tener en cuenta, es la calidad del servicio, en el cual uno de los elementos más importantes es la continuidad del mismo, siendo una preocupación la pérdida total del suministro y en un segundo lugar se encuentran las pérdidas parciales del servicio.

Para dar soporte a este problema, los sistemas inteligentes son una importante alternativa a explorar. Los sistemas basados en técnicas de la Inteligencia Artificial y en particular la tecnología de los sistemas multi-agentes, son una herramienta valiosa a la hora de desarrollar sistemas para problemas reales complejos, de carácter distribuido y donde se espera que sus componentes tengan cierta autonomía. Existen diversos trabajos enfocados al desarrollo de sistemas inteligentes en este dominio de aplicación, por ejemplo [14] y [15].

[1]Comprenden a los grandes consumidores de energía eléctrica, tales como las industrias del acero, químicas, papel, etc.; que generalmente reciben el suministro eléctrico en alta o media tensión.

1.1. Dominio de aplicación

Un SEP es un conjunto de equipos que permiten energizar cargas en forma segura y confiable, en distintos niveles de tensión, ubicados generalmente en diferentes lugares físicos.

Entre los componentes que se encuentran en un SEP, podemos mencionar principalmente a Protecciones Eléctricas (Relays), Interruptores, Transformadores, Barras, Generadores, Líneas y Cargas.

Las protecciones eléctricas tienen un rol fundamental en el proceso de adquisición y análisis de datos. Una protección eléctrica (también llamada *relé*) es un dispositivo que sensa variables de redes, para procesarlas mediante algoritmos adecuados. A su vez ordena acciones frente a situaciones anormales. Existen distintos tipos de protecciones que se clasifican en base a su función. Para el trabajo realizado se considera principalmente el relé denominado *de máxima corriente y direccional*. Cada relé tiene asignada una "corriente de seteo" (I_c). I_c es superior a la máxima corriente que puede circular por el relé en condiciones normales en al menos un 30 %, para tener en cuenta errores de calibración. Se pueden definir las principales señales provenientes de un relé de la siguiente manera:

$$S_1 = \begin{cases} 1 & \text{si } I \geq I_c \\ 0 & \text{si } I < I_c \end{cases}$$

$$S_2 = \begin{cases} 1 & \text{si } I \text{ va de Barra a Línea} \\ 0 & \text{en } I \text{ va de Línea a Barra} \end{cases}$$

$$S_3 = \begin{cases} 1 & \text{si el Relé presenta alguna falla interna} \\ 0 & \text{si no} \end{cases}$$

Si la corriente (I) que circula es mayor (o igual) que I_c, entonces el relé se *excita* (también suele decirse que opera). Esta información se obtiene mediante S_1. Mediante S_2 obtenemos la dirección en que circula la corriente y mediante S_3 podemos saber si el relé presenta alguna falla interna (por motivo desconocido). Otros tipos de protecciones que consideramos son las denominadas *protecciones propias de transformador (PPT)*. Estas presentan señales de excitación y falla-relé (falla interna). También ordenan disparos.

Las protecciones de máxima corriente, tienen un tiempo de seteo t_a. Si la protección se encuentra en estado de excitación durante el tiempo t_a, inmediatamente ordena la apertura del interruptor asociado (se dice que ordena disparo o "trip").

Un interruptor (de potencia) es un dispositivo utilizado para desconectar una carga o una parte del sistema eléctrico, tanto en condiciones de operación normal como en condición de cortocircuito. La operación de un interruptor puede ser automática o manual, accionada por la señal de un relé encargado de vigilar la correcta operación del sistema eléctrico donde está conectado o de un operador a distancia. Cada interruptor suele estar asociado a un relé.

Las barras pueden ser consideradas como nodos del SEP. Son puntos de maniobra pues a ella concurren las líneas, se conectan los transformadores y las cargas. Las cargas son elementos que transforman la energía eléctrica en otra forma de energía: mecánica, calórica, etc; pueden ser por ejemplo: motores, hornos, equipos de iluminación, etc. Las líneas cumplen el rol de conductores y transmisores de la energía. Un transformador es un equipo que permite aumentar o disminuir la tensión en un circuito eléctrico. Un generador es una fuente productora de energía eléctrica.

Es muy importante mantener el suministro de energía en forma adecuada, pues los daños ocasionados pueden ser de costos muy elevados e incluso se puede ver afectada la vida de seres humanos, si consideramos por ejemplo un sistema de ventilación. Resulta necesaria entonces la existencia de sistemas de control de SEPs.

1.2. Problemática actual

En la actualidad se cuenta con sistemas SCADA para la adquisición de datos, aplicaciones aisladas que realizan análisis de aspectos específicos y también software para soporte de decisión.

La complejidad actual de los SEPs radica en que no es posible disponer de una regulación (seteo) única para todas las condiciones operativas factibles. En las redes actuales se pueden presentar operaciones no consideradas previamente debido a servicios prestados bajo severidades extremas motivadas por salidas de servicio de una parte de las instalaciones y ante la necesidad de continuar con el suministro, se alteran las condiciones preestablecidas. Entre ellas se pueden citar sobrecargas en líneas que llevan a desconexiones en cascada.

Es necesario que para lograr un funcionamiento adecuado de las protecciones eléctricas, se produzca una variación en su seteo en forma automática (relé adaptivo), teniendo en cuenta estados de operación de elementos adyacentes y lejanos [2].

Con el fin de desarrollar un sistema de control acorde a las necesidades vigentes se utilizaron en este proyecto herramientas de la Inteligencia Artificial, principalmente se hizo uso de tecnología multi-agente y de Sistemas de Razonamiento Procedural (PRS) [12]. El resultado de este trabajo es el diseño de un Sistema Inteligente de Control (SIC) aplicado a redes eléctricas y la implementación de dos de sus agentes (el agente Detección y Aislación de Fallas (DAF) y el agente Back-Up) utilizando el modelo BDI. Esta presentación tiene la siguiente estructura: en la sección 2 se describen los requerimientos que debe verificar el SIC. En las secciones 3 y 4 se desarrollan respectivamente la arquitectura del SIC y el diseño de sus agentes. Finalmente, las secciones 5 y 6 corresponden a observaciones sobre la experimentación y conclusiones.

2. REQUERIMIENTOS

Por medio de un trabajo de Ingeniería de Conocimiento se realizó la extracción de requerimientos para el sistema planteado. Mediante sucesivas reuniones con el experto en el dominio de aplicación [2] se logró refinar y comprender los objetivos del mismo.

En primera instancia se identifica el siguiente objetivo general del sistema de control: *actuar ante una falla, tratando de mantener la integridad del mayor área posible bajo condiciones de calidad adecuadas*. Luego, se puede descomponer el objetivo global en las siguientes tareas o subojetivos:

- detectar fallas;
- aislar fallas;
- respaldar componentes que presenten falla interna (protecciones e interruptores);
- restaurar el servicio a zonas desenergizadas y
- mantener un modelo del SEP.

En este trabajo se centra la atención en las tres primeras tareas. También debe considerarse un importante requerimiento temporal. Los tiempos de actuación en caso de existir una falla en el SEP deben ser del orden de 100 milisegundos. Si bien el presente trabajo no pretende necesariamente que la implementación verifique este requerimiento dado que se trata de un prototipo, debe ser considerado, ya que una implementación final sí debería verificarlo.

[2]Ing. Luis A. Krapf , Escuela de Ingeniería Eléctrica, Facultad de Cs. Exactas, Ingeniería y Agrimensura, UNR

Un aspecto que también se debe tener en cuenta es la capacidad de adquirir datos provenientes del SEP, realizar algún tipo de preprocesamiento (e.g., filtrado[3]) y luego transmitir eventos significativos a las entidades interesadas.

Otro requerimiento, no funcional, involucra la posibilidad o facilidad para integrar sistemas de análisis preexistentes o "legacy". El sistema a desarrollar deber permitir una fácil integración.

Para continuar con el proceso de adquisición del conocimiento, se obtuvieron una serie de escenarios, también llamados *casos de uso*. Cada caso de uso representa una situación de falla y determina las acciones necesarias que deben efectuarse correspondientemente. Esto permite realizar un *refinamiento en los requerimientos*. Por cada falla analizada se genera un caso de uso, el cual a su vez es representante de un conjunto de situaciones que deben ser tratados de forma equivalente. Esta agrupación se debe sencillamente que dependiendo de la configuración de cada protección se puede excitar un mayor o menor número de protecciones. Las protecciones que deben tener un comportamiento común para toda instancia de una clase de casos de uso son aquellas más próximas a la falla, salvo en casos de incertidumbre en los que estas protecciones pueden no dar indicaciones (considerando de esta manera fallas internas, funcionamientos anómalos o errores de calibración).

A continuación se describe un escenario a modo de ejemplo. La situación que se desea detectar se describe en forma gráfica (ver Fig. 1). Para simplificar la notación, no se introducen los nombres de los interruptores. Se asume en términos generales que el relé R_j está asociado al interruptor I_j. Además pueden aparecer flechas debajo de algunos componentes. Estas corresponden a indicaciones (o señales) del sentido de la corriente provistas por los relés. Si bien un relé puede dar este tipo de indicación sin estar excitado, debe asumirse que lo está (también con el fin de simplificar la notación).

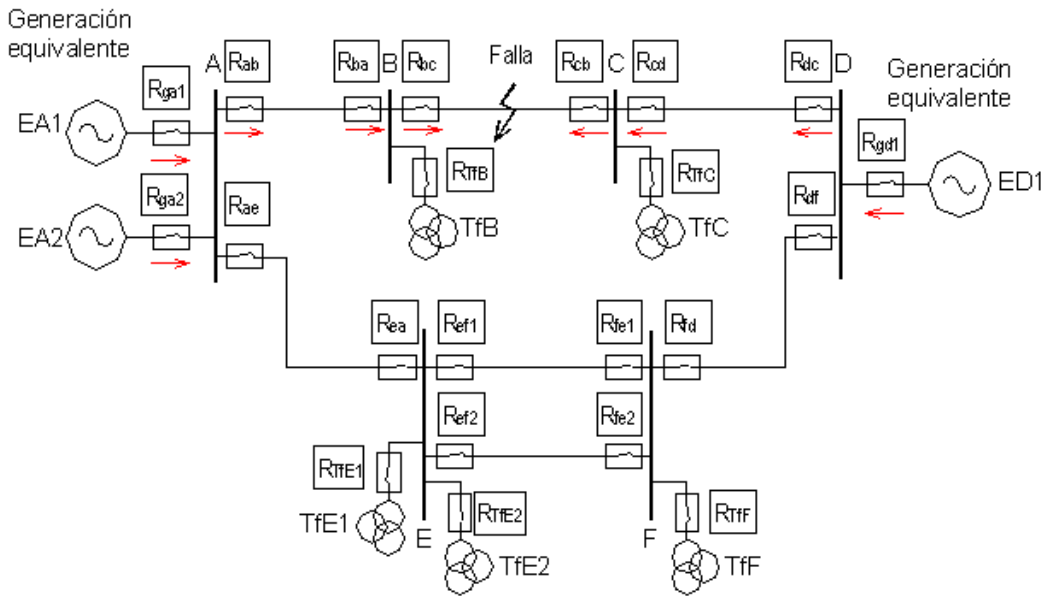


Figura 1: Caso 1. Falla en LINEA

En el caso ilustrado en la figura 1 se presenta la falla en una línea. Las **acciones** necesarias para resolverlo son:

- disparar los interruptores I_{bc} y I_{cb}.

[3]evitando por ejemplo, el envío de señales ante una intermitencia en las mismas

- si no abre I_{bc} se debe disparar I_{ba} y I_{ab}. También deben desconectarse las cargas involucradas, o sea, abrir I_{Tfb} y las cargas colgadas del transformador (TfB) deben desconectarse recursivamente (empezando por la más cercana).
- análogamente, si no abre I_{cb} se debe disparar I_{cd} y I_{dc}. También deben desconectarse las cargas involucradas, o sea, abrir I_{TfC}. Las cargas colgadas del transformador (TfC) deben desconectarse recursivamente (empezando por la más cercana).
- se deben impedir (bloquear) los disparos de los demás relés que se excitaron.

En este trabajo concretamente se identificaron nueve casos de uso, abarcando entre ellos existencia de incertidumbre. Se presenta incertidumbre cuando la información recolectada es incompleta para alcanzar una solución (por ejemplo el caso en que una protección no envíe señal cuando en realidad debiera hacerlo) o cuando la información obtenida es errónea y genera inconsistencias (por ejemplo el envío de señales incorrectas, como ser la dirección de corriente opuesta a la real) debiéndose esto a fallas internas o funcionamientos anormales.

2.1. Características del problema

El problema en el cual nos concentramos tiene algunas características importantes a considerar. Los datos utilizados para el análisis del SEP, se encuentran distribuidos geográficamente, puesto que la información puede ser recolectada de diferentes estaciones remotas. Estos datos tienen a su vez una naturaleza altamente dinámica. Por otro lado, la complejidad del problema puede ser atacada mediante tareas independientes. Estas características, claramente nos conducen a que la aplicación de la tecnología de sistemas Multi-Agentes [10] es altamente adecuada para resolver un problema como el propuesto.

3. ARQUITECTURA MULTI-AGENTE

Durante los últimos años se han desarrollado varias metodologías para el análisis y diseño orientado a agentes (AOSE). Ejemplos de esto son DESIRE [7], Gaia [11], Prometheus [8] y para el desarollo de agentes BDI [1]. Si bien hay similitudes, cada metodología tiene aspectos muy interesantes, por lo que se extrajeron conceptos de las mismas para avanzar hacia la etapa diseño.

3.1. De los Requerimientos hacia la Arquitectura

El primer paso seguido es la *descomposición de tareas* (según la metodología DESIRE [7]). Este proceso tiene como fin la obtención de tareas o funcionalidades requeridas.

Las funcionalidades deben ser lo más simples posible de manera de tratar con un único aspecto o comportamiento esperado del sistema. Por lo tanto podemos identificar las siguientes funcionalidades del sistema: *identificar fallas*; *bloquear disparos innecesarios*; *aislar fallas identificadas*; *administrar eventos*; *respaldar componentes que presenten falla (Back-Up)* y *restaurar el servicio*.

Al definir una funcionalidad también es importante definir la información que requiere y la información que produce. Para describir o especificar una funcionalidad se utilizan descriptores que contienen el *nombre* de la misma, una *descripción* informal breve, una lista de *acciones*, una lista de *percepciones*, *datos usados* y *producidos*, y una breve descripción de las interacciones con otras funcionalidades.

Esta forma de identificar funcionalidades, da la posibilidad de estudiar la interacción entre ellas, lo cual es necesario para el siguiente paso. Este consiste en identificar cuáles serán los agentes que van a existir en el sistema.

Veamos la descripción de una funcionalidad:

Nombre: Detector de fallas.
Descripción: Determina la presencia de falla, su localización y su tipo.
Acciones: -.
Percepción. Estado de protecciones.
Acceso a Datos: Lectura del estado de protecciones, topología de red. Escritura de datos con la región donde se encuentra la falla y su clasificación.
Interacciones: Provee la región generada a las funcionalidades Bloqueo y Aislación. A esta última también le provee la clase y locación de falla.

En el diseño de un sistema Multi-Agente, resulta fundamental definir los agentes que lo componen, así como también las interacciones entre ellos. Se puede considerar a un agente como una combinación de funcionalidades. Luego, se puede determinar qué agentes existirán evaluando diferentes combinaciones según criterios provenientes del diseño orientado a objetos y reutilizados en AOSE. Se pueden asignar funcionalidades a agentes de modo de encontrar una configuración en la que se consiga un bajo nivel de acoplamiento y un gran nivel de cohesión. Por ejemplo, si distintas funcionalidades usan los mismos datos entonces es una indicación de que se deben agrupar dado que habrá una gran interacción entre ellas. Si distintas funcionalidades están claramente no relacionadas es una razón para no agruparlas, así como también si existen en distintas plataformas de hardware.

Utilizamos un diagrama de interacción entre funcionalidades (ver figura 2) para mostrar las funcionalidades (rectángulos), los datos (elipses) y conexiones entre ellos (flechas). Una flecha desde una funcionalidad hacia datos indica que la funcionalidad produce los datos.

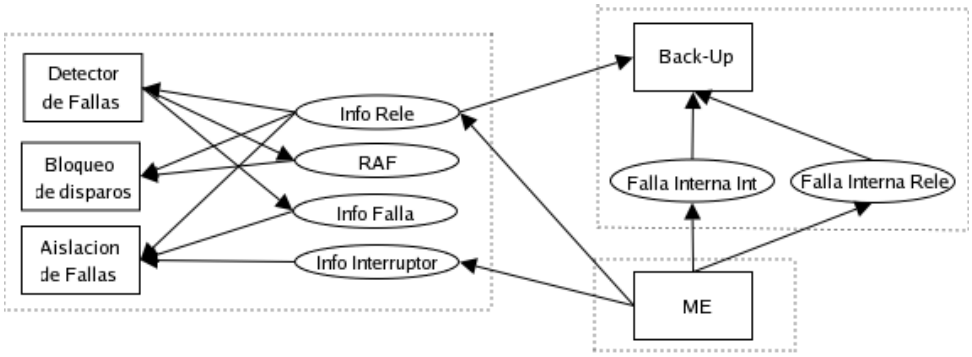


Figura 2: Interacción entre funcionalidades

Del análisis de las funcionalidades antes identificadas y del diagrama de interacción resulta claro y natural agrupar las funcionalidades de forma tal que se obtienen los siguientes agentes:

- Detección y Aislación de fallas (DAF)
- Back-Up (BK)
- Manejador de Eventos (ME)
- Reposición de Servicio (RS)

Además de los mencionados consideramos dos tipos de agente, cuya representación está motivada por correspondencia directa con elementos muy importantes para el dominio de aplicación:

- Agente de Protección (AP)
- Agente de Interrupción (AI)

En el presente trabajo no se describe el diseño de los agentes RS y ME. Sí se consideran aspectos relacionados a la interaccón con el agente ME, dado que éste constituye el nexo entre los agentes DAF y Back-Up y el sistema eléctrico subyacente (SEP). El Agente de Protección es un tipo de agente que representa una protección eléctrica. Puede actuar autónamente para proteger dispositivos del SEP. Consiste principalmente de reglas, lógica de relés, y características de operación. El Agente de Interrupción representa el comportamiento de un interruptor.

A continuación se describe un agente tal como aparece en *la guía de agentes*, siguiendo el estilo de notación empleado en Prometheus [8].

Nombre: Agente de Detección y Aislación de Fallas (DAF)
Descripción: Analiza los eventos provenientes de una capa inferior con el objetivo de detectar y encontrar con la mayor precisión posible la locación de una falla eventual para luego poder aislarla y evitar una desenergización innecesaria.
Cardinalidad: 1
Funcionalidades incluidas: Identificar región de falla, bloquear disparos innecesarios, aislar fallas
Lee datos: Estado de los componentes (modelo propio)
Escribe datos: Estado de los componentes (modelo propio)
Interactúa con: ME (para obtener eventos de interés y ordenar acciones), Back-Up (obtener correcciones en la información sobre componentes con falla interna)

3.2. Interacción entre Agentes

Tan importante como definir los agentes que forman el sistema, es describir la interacción que habrá entre ellos. La figura 3 muestra un diagrama de interacción entre los agentes propuestos. Este diagrama simplemente vincula agentes que tienen algún tipo de interacción, ya sea mediante datos compartidos o mensajes. La existencia de múltiples agentes de un mismo tipo se representa en el diagrama mediante rectángulos superpuestos. Resulta útil este diagrama para considerar el nivel de acoplamiento entre agentes.

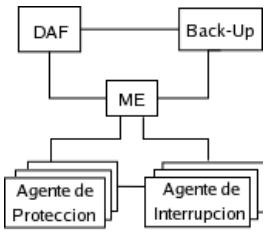


Figura 3: Interacción básica entre Agentes

Mediante un mecanismo de suscripción, el Agente Manejador de Eventos (ME), mantiene registros de cuáles son los eventos de interés para cada agente existente en el sistema y es su función enviarlos. Por lo tanto es importante, definir cuáles serán los eventos generados como resultado de la información obtenida del entorno (SEP). Los eventos generados serán los percibidos por los agentes

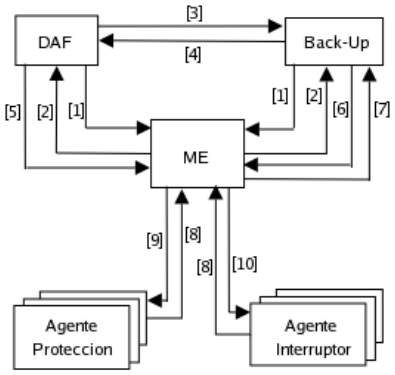


Figura 4: Diagrama de Colaboración.

Id	*Descripción del mensaje*
1	Suscripción por eventos de interés.
2	Informe a los agentes suscriptos de nuevo evento de interés.
3	Suscripción por corrección de lecturas (de relés con falla).
4	Informe a los agentes suscriptos de correcciones.
5	Orden de acción (por ejemplo disparo).
6	Solicitud de información (por ejemplo: estado de un relé).
7	Informe con respuesta a solicitud.
8	Informe con señales o datos propios de componentes eléctricos.
9	Solicitud de bloqueo.
10	Solicitud de disparo o cierre.

Figura 5: Mensajes entre agentes.

de análisis como por ejemplo, los agentes DAF y Back-Up. También se deben definir los eventos generados por estos últimos. Los agentes se comunican entre sí mediante mensajes. Un lenguaje para la comunicación de agentes (ACL) es un medio adecuado de encapsulación de mensajes para su transporte [6].

La interacción entre agentes puede ser capturada mediante distintas representaciones. Una representación posible está dada por "diagramas de colaboración", otra esta dada por "diagramas secuenciales". Los diagramas secuenciales hacen énfasis en la secuencia cronológica de la comunicación mientras que los diagramas de colaboración hacen énfasis en las asociaciones entre agentes.

La figura 4 y el cuadro de la figura 5 ejemplifican cómo es la interacción entre los agentes. Los números de referencia que se encuentran sobre los líneas de asociación dan una orden a una secuencia general de interacción posible, que está descripta en la tabla adyacente.

Mediante un diagrama secuencial (ver figura 6) se puede permitir una mejor visualización de la forma en que interactúan o colaboran los agentes en el caso particular en que el agente DAF (suscripto al servicio de corrección de lecturas provista por el agente Back-Up) ordena la apertura de un interruptor con falla interna. En este tipo de diagramas, se puede leer la secuencia temporizada de interacción desde arriba hacia abajo (o sea, el tiempo se incrementa hacia abajo). En este caso se simplificó la notación agrupándose a los Agentes de Protección y Agentes de Interrupción dentro de SEP.

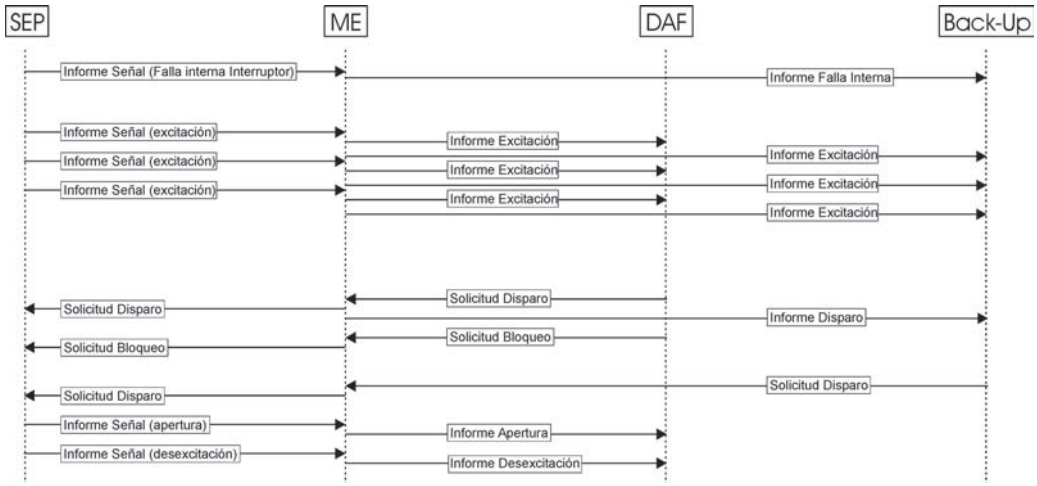


Figura 6: Diagrama secuencial. Presencia de falla interna en interruptor.

A partir de la arquitectura de este sistema multiagente denominado SIC, en la próxima sección se avanza en el diseño e implementación de dos de sus agentes: el agente DAF y el agente Back-Up.

4. AGENTES DAF Y BACK-UP

En esta sección se describen los puntos relevantes del diseño e implementación de los agentes DAF y Back-Up. Del proceso de adquisición del conocimiento resulta natural pensar en un conjunto o jerarquía de procedimientos para determinar cómo estos agentes llevarán a cabo sus tareas. Este proceso deliberativo para decidir qué acción deberá tomar el agente en cada caso, se puede trasladar a la toma de decisión en torno a las intenciones en un sistema intencional. Además, los agentes DAF y Back-Up deben combinar acciones de naturaleza reactiva y proactiva, por lo cual se decide modelizarlos bajo el paradigma BDI, utilizando una arquitectura PRS.

La arquitectura PRS[4] (originalmente desarrollada Georgeff y Lansky [12]) fue quizás la primera arquitectura basada en el paradigma BDI y se ha convertido en una de las más conocidas. Ha sido utilizada en varias aplicaciones reales (e.g. [9], [13]). Entre las ventajas principales de PRS se destacan: la representación de planes y procedimientos, la posibilidad de utilizar planes parciales, comportamiento Reactivo y Proactivo, y la capacidad de Meta-Razonamiento, es decir planes que razonan o gestionan otros planes.

Durante los últimos años se ha construido un gran número de plataformas o entornos para el desarrollo de agentes y en particular de PRS. De las plataformas existentes se buscaron aquellas de código abierto[5] y escritas fundamentalmente en lenguaje C. Esta última preferencia se basa en que el código C ha mostrado no tener inconvenientes bajo restricciones de tiempo real duro. La plataforma de desarrollo elegida resultó ser OpenPRS (OPRS) [4], una versión de código abierto de PRS [5].

4.1. Sistema de Razonamiento Procedural (PRS)

Un Sistema de Razonamiento Procedural es un conjunto de herramientas y métodos para la representación y ejecución de planes y procedimientos. Una arquitectura PRS consiste básicamente en: (1) una base de datos con las *creencias* actuales sobre el entorno; (2) un conjunto de *objetivos o deseeos* actuales a alcanzar; (3) una librería de planes o *procedimientos*, que describen secuencias particulares de acciones y pruebas que pueden realizarse para alcanzar ciertos objetivos o para reaccionar ante ciertas situaciones; y (4) una *estructura de intenciones*, que consiste de un conjunto ordenado (parcialmente) de todos los planes elegidos para ejecución. Estos componentes y sus interacciones se ilustran en la figura 7.

Un *intérprete* (mecanismo de inferencia) manipula estos componentes. Recibe nuevos eventos y objetivos internos; selecciona un plan apropiado teniendo en cuenta los nuevos eventos, objetivos y creencias; ubica el plan dentro de la estructura de intenciones (grafo); elije un plan (intención) en la estructura y finalmente ejecuta un paso del plan activo. Esto puede resultar en una acción primitiva o en la formulación de un nuevo objetivo. El sistema interactúa con el entorno a través de su base de datos y de las acciones básicas o primitivas.

Los algoritmos utilizados y el ciclo principal que presenta el Intérprete de OPRS, considerando algunos supuestos razonables, permiten garantizar una cota superior sobre el tiempo de reacción. Por ejemplo, se debe considerar un máximo en la frecuencia de llegada de eventos y se debe verificar que la cardinalidad del conjunto de planes aplicables se decrezca monótonamente. Un análisis detallado de esto puede encontrarse en [3].

[4] Procedural Reasoning System

[5] Open Source

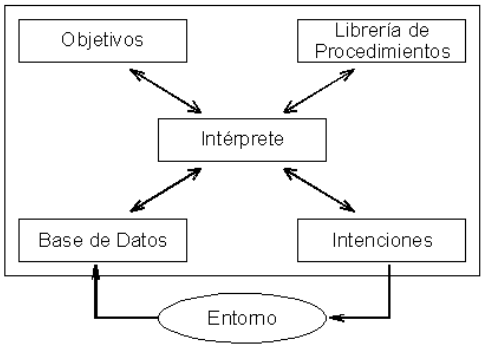


Figura 7: PRS. Una arquitectura basada en el paradigma BDI

4.2. Diseño

Para diseñar un agente con arquitectura PRS es necesario y fundamental especificar sus *creencias* y también su *librería de planes*. Con estos componentes, el *intérprete* (cuya implementación está provista por la plataforma OPRS) genera dinámicamente los componentes de *objetivos* e *intenciones*.

Las creencias de un agente son básicamente una base de datos de formato simple, la cual consiste en una lista de expresiones (predicados). Un ejemplo de archivo que representa parte de la base de datos del agente DAF en un instante dado es:

```
(
(EXCITED R3 1)
(EXCITED R4 1)
(POSITION I3 CLOSED)
(POSITION I4 CLOSED)
(ASSOCIATED-RI R5 I5)
(ASSOCIATED-RI R3 I3)
)
```

Podemos observar entonces que el agente DAF cree o tiene conocimiento de los siguientes hechos: los relés $R3$ y $R4$ se encuentran excitados, los interruptores $I3$ e $I4$ están cerrados y los relés $R3$ y $R5$ están asociados a los interruptores $I3$ e $I5$ respectivamente.

La plataforma OPRS permite especificar los planes mediante una herramienta gráfica. Un ejemplo de plan generado con esta herramienta es la figura 8.

Por último, se puede considerar a cada agente como un núcleo (OPRS Kernel) compuesto por un conjunto de planes y por una base de conocimientos.

Siguiendo la metodología Prometheus [8], una forma de documentar elegantemente el diseño de un agente, consiste en determinar sus *capacidades*. Estas pueden pensarse como módulos dentro del agente y pueden estar anidadas dentro de otras capacidades. En el nivel inferior de anidamiento las capacidades se describen en términos de *eventos internos*, *planes* y *estructuras de datos*. Para fines de documentación, cada capacidad puede ser definida mediante un *Descriptor de Capacidad*, el cual brinda información sobre la interfaz de la capacidad (los eventos que sirven como entrada y los eventos producidos), información sobre interacción con otras capacidades, las capacidades incluidas, y referencias a datos de lectura y escritura.

Las funcionalidades descriptas en la sección 3.1 forman un conjunto inicial de capacidades, el cual puede ser refinado si así se quisiera. A continución se describe una de las capacidades del agente DAF a modo de ejemplo y también uno de los planes que comprende esta capacidad (figura 8).

Nombre: Aislación.

Descripción: Aisla la falla detectada, con posible manejo de incertidumbre.

Eventos de salida: Ordena disparos (apertura de interruptores).
Eventos de entrada: Estado de las protecciones.
Capacidades incluidas: -
Planes: AISLAR-F-B, AISLAR-F-L, AISLAR-F-T, AISLAR-F-TB, APERTURA-CNTRL, APERTURA-REC, APERTURA-VEC, APERTURA-MUL-VEC, ASSOCIATED-LRI, INIT-AIS, INIT-AIS-INCERT, INIT-AIS-INCERT2, OBTENER-RAF-TOTAL
Acceso a Datos: Lectura del estado de las protecciones, topología de red.
Interacciones: Necesita la información sobre la falla (clase y locación) producida por Detector de Fallas.

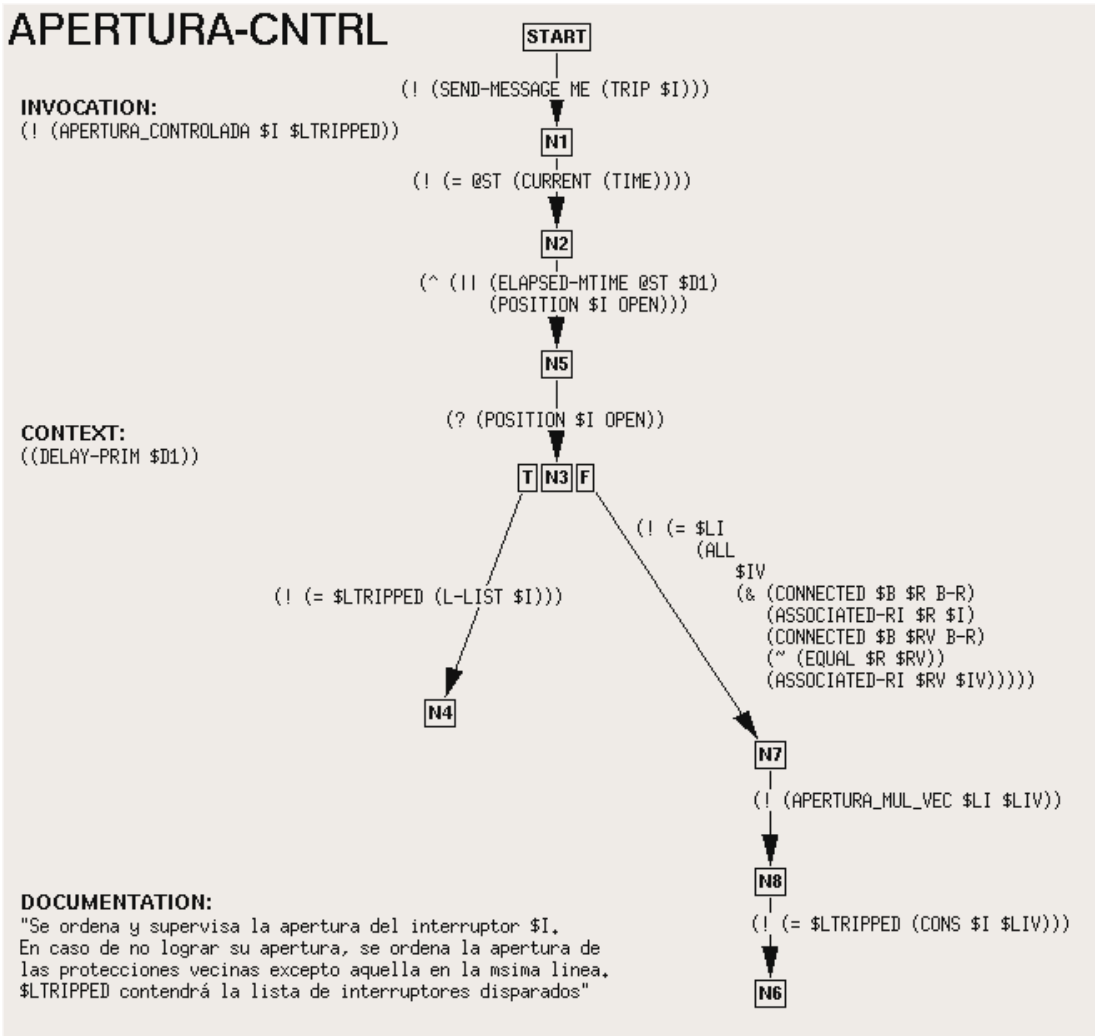


Figura 8: Un plan para controlar la apertura de un interruptor

5. EXPERIMENTACIÓN Y SIMULACIÓN

Se consideraron dos modalidades de prueba. En la primera se supone que el agente DAF parte de un estado en que cuenta con la información correspondiente a situaciones específicas de interés. Basta con enviarle un mensaje de **excitación** del relé que dará inicio a la detección. En la segunda se considera que el agente DAF no tiene la información que describe cada situación sino que el agente

ME envía la información simulando un comportamiento posible del SEP para alcanzar dicha situación de prueba. Todos los casos tratados tuvieron la respuesta esperada.

Por último se destaca que la herramienta OPRS utilizada brinda una interfaz gráfica (X-OPRS) la cual permite interactuar con cada agente, analizar su respuesta ante estímulos manuales y también observar la evolución de las estructuras cognitivas que lo representan. Esta herramienta visual, disminuye notoriamente la performance (tiempos de respuesta), pero esto carece de importancia puesto que la utilización de la interfaz gráfica es opcional y sus fines consisten en permitir una mejor concepción y comprensión del comportamiento del SIC.

6. CONCLUSIONES

En este trabajo se analizó y describió la arquitectura general de un Sistema Inteligente de Control (SIC) para SEPs y la arquitectura propia de dos agentes, el agente de Detección y Aislación de Fallas y el agente de Back-Up. Como resultado se obtiene una arquitectura sumamente flexible, ya que permite realizar cambios en cualquiera de los componentes estudiados en este trabajo, sin afectar de manera importante al resto. Esto vale para el caso del sistema multiagente y también a nivel de agente.

A nivel de sistema multi-agente, se cuenta con características importantes como por ejemplo: cooperación de agentes autónomos, lo cual permite el manejo de tareas independientemente, clara interacción entre los agentes a través de mensajes y la posibilidad de trabajar con información de manera descentralizada.

A nivel de diseño de los agentes, la arquitectura PRS hace que los agentes desarrollados posean cualidades muy ventajosas. En primer lugar su incrementabilidad, pues permite muy fácilmente satisfacer nuevos requerimientos o cambios en los mismos. Además los procedimientos gozan de una representación muy ventajosa ya que pueden ser parciales, delegando al intérprete la responsabilidad de satisfacer los subobjetivos que estos requieran alcanzar y permitiendo un refinamiento en las tareas. La noción de objetivo es fuerte en PRS, dado que el intérprete considera todos los procedimientos que unifiquen con él para alcanzarlo, antes de considerar que falla. Por otra parte, la combinación de comportamiento proactivo y reactivo (en tiempo acotado) es fundamental para este desarrollo, pues se necesita comportamiento reactivo cada vez que el SEP de indicaciones de fallas potenciales y se requiere comportamiento proactivo para realizar por ejemplo, tareas de detección bajo incertidumbre. Otra ventaja es la posibilidad de controlar parte del ciclo principal mediante meta-razonamiento, utilizando meta-planes. Por último, se destaca que como resultado final de la etapa de diseño se obtiene directamente la implementación del sistema.

REFERENCIAS

[1] Casali A., Godo L., and Sierra C. A methodology to engineer graded bdi agents. In *WASI - CACIC Workshop.XII Congreso Argentino de Ciencias de la Computación*, 2006.

[2] Phadke A.G. and Horowitz S.H. Adaptive relaying. *IEEE Computer Applications in Power*, 3(3):47–51, 1990.

[3] Ingrand F. and Coutance V. Real-time reasoning using procedural reasoning. Technical report 93-104, LAAS/CNRS, Toulouse, France, 1993.

[4] Ingrand F.F. OPRS development environment, 2004. http://softs.laas.fr/openrobots/php/download.php/openprs-1.0b1-doc.pdf.

[5] Ingrand F.F., Georgeff M.P., and Rao A.S. An architecture for real-time reasoning and system control. *IEEE Expert: Intelligent Systems and Their Applications*, 7(6):34–44, 1992.

[6] FIPA. *FIPA ACL Message Structure Specification*. FIPA, 2001.

[7] Brazier F.M.T, Dunin-Keplicz B.M., Jennings N.R., and Treur J. DESIRE: Modelling multi-agent systems in a compositional formal framework. *Int Journal of Cooperative Information Systems*, 6(1):67–94, 1997.

[8] Padgham L. and Winikoff M. Prometheus: A methodology for developing intelligent agents. In *AAMAS 2002*, Bologna, Italy, July 2002.

[9] Ljungberg M. and Lucas A. The OASIS air-traffic management system. In *PRICAI '92*, Seoul, Korea, 1992.

[10] Wooldridge M. *An Introduction to Multi-Agent Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[11] Wooldridge M., Jennings N.R., and Kinny D. The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.

[12] Georgeff M.P. and Lansky A.L. Reactive reasoning and planning. In *Proc. of AAAI-87*, pages 677–682, Seattle, WA, 1987.

[13] Georgeff M.P. and Ingrand F.F. Monitoring and control of spacecraft systems using procedural reasoning. Technical Report 03, Australian Artificial Intelligence Institute, Melbourne, Australia, November 1989.

[14] McArthur S.D.J., Davidson E.M., Hossack J.A., and McDonald J.R. Automating power system fault diagnosis through multi-agent system technology. In *HICSS*, 2004.

[15] Thorp J.S. Giovanini R. Birman K. Coury D. Wang X.R., Hopkinson K.M. Developing an agent-based backup protection system for transmission networks. *Power Systems and Communications Infrastructures for the Future, Beijing, September 2002*.

Solving Constrained Optimization using a T-Cell Artificial Immune System

Victoria S. Aragón, Susana C. Esquivel *
Universidad Nacional de San Luis
Ejército de los Andes 950
(5700) San Luis - Argentina
{vsaragon, esquivel}@unsl.edu.ar

and

Carlos A. Coello Coello †
Electrical Eng. Department, Computer Science Dept.
Av. IPN No. 2508, Col. San Pedro Zacatenco
México D.F. 07300, MÉXICO
ccoello@cs.cinvestav.mx

Abstract

In this paper, we present a novel model of an artificial immune system (AIS), based on the process that suffers the T-Cell. The proposed model is used for solving constrained (numerical) optimization problems. The model operates on three populations: Virgins, Effectors and Memory. Each of them has a different role. Also, the model dynamically adapts the tolerance factor in order to improve the exploration capabilities of the algorithm. We also develop a new mutation operator which incorporates knowledge of the problem. We validate our proposed approach with a set of test functions taken from the specialized literature and we compare our results with respect to Stochastic Ranking (which is an approach representative of the state-of-the-art in the area) and with respect to an AIS previously proposed.

Keywords: Artificial Immune System, Constrained Optimization Problem.

Resumen

En este trabajo, se presenta un nuevo modelo de Sistema Inmune Artificial (SIA), basado en el proceso que sufren las células T. El modelo propuesto se usa para resolver problemas de optimización (numéricos) restringidos. El modelo trabaja sobre tres poblaciones: Vírgenes, Efectoras y de Memoria. Cada una de ellas tiene un rol diferente. Además, el modelo adapta dinamicamente el factor de tolerancia para mejorar las capacidades de exploración del algortimo. Se desarrolló un nuevo operador de mutación el cual incorpora conocimiento del problema. El modelo fue validado con un conjunto de funciones de prueba tomado de la literatura especializada y se compararon los resultados con respecto a Stochastic Ranking (el cual es un enfoque representativo del estado del arte en el area) y con respecto a un SIA propuesto previamente.

Palabras claves: Sistema Inmune Artificial, Problemas de Optimización Restringidos.

*Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
†CINVESTAV-IPN (Evolutionary Computation Group)

1 INTRODUCTION

In many real-world problems, the decision variables are subject to a set of constraints, and the search has to be bounded accordingly. Constrained optimization problems are very common, for example, in engineering applications, and therefore the importance of being able to deal with them efficiently.

Many bio-inspired algorithms (particularly evolutionary algorithms) have been very successful in the solution of a wide variety of optimization problems [30]. But, when they are used to solve constrained optimization problems, they need a special method to incorporate the problem's constraints into their fitness function. Evolutionary algorithms (EAs) often use exterior penalty functions in order to do this [26]. However, penalty functions require the definition of accurate penalty factors and performance is highly dependent on them.

Recently, several researchers have proposed novel constraint-handling techniques for EAs [3, 19, 24]. These approaches have been able to outperform penalty functions and can handle all types of constraints (linear, nonlinear, equality, inequality).

The main motivation of the work presented in this paper is to explore the capabilities of a new AIS model in the context of constrained global optimization. The proposed model is based on the process that suffers the T-Cell. We also propose a dynamic tolerance factor and several mutation operators that allow us to deal with different types of constraints. The remainder of the paper is organized as follows. In Section 2, we define the problem we want to solve. Section 3 describes some previous work. In Section 4, we introduce and describe our proposed model. In Section 5, we present our experimental setup. In Section 6, our results are presented and they are discussed. Finally, in Section 7, we present our conclusions and some possible paths for future work.

2 STATEMENT OF THE PROBLEM

We are interested in solving the general nonlinear programming problem which is defined as follows: Find $\vec{x} = (x_1, \ldots, x_n)$ which optimizes $f(x_1, \ldots, x_n)$ subject to:

$$h_i(x_1, \ldots, x_n) = 0 \; i = 1, \ldots, l$$
$$g_j(x_1, \ldots, x_n) \leq 0 \; j = 1, \ldots, p$$

where $(x_1, \ldots, x_n)$ is the vector of solutions (or decision variables), l is the number of equality constraints and p is the number of inequality constraints (in both cases, constraints could be linear or nonlinear).

3 PREVIOUS WORK

According to [10] the main models of Artificial Immune System are: Negative Selection, Clonal Selection and Immune Network Models. They are briefly described next.

Forrest et al. [25] proposed the Negative Selection model for detection of changes. This model is based on the discrimination principle that the immune system adopts to distinguish between self and nonself. This model generates random detectors and discards the detectors that are unable of recognizing themselves. Thus, it maintains the detectors that identify any nonself. It performs a probabilistic detection and it is robust because it searchess any foreign action instead of a particular action. Typical applications of negative selection [10] include those reported in [25, 11, 7], among others.

The Immune Network Model was proposed by Jerne [17], and it is a mathematical model of the immune system. In this case, the dynamics of the lymphocytes are simulated by differential equations. This model assumes that lymphocytes are an interconnected network. Several models have been derived from it [12, 1]. Typical applications are [10]: detection of gene promoter sequences [13], data mining [14], diagnosis [15] and cluster analysis [16, 27].

Clonal Selection is based on the way in which both B-cells and T-cells adapt in order to match and kill the foreign cells [10]. Clonal Selection involves: 1) the AIS' ability to adapt its B-cells to new types of antigens and 2) the affinity maturation by hypermutation. CLONALG proposed by Nunes de Castro and Von Zuben [22, 23] was originally used to solve pattern recognition and multimodal optimization problems, and there are a few extensions of this algorithm for constrained optimization. CLONALG works in the following way: first, it creates a random population of antibodies, it sorts it according to some fitness function, it clones them, it mutates each clone, it selects the fittest antibodies and clones it and replaces the worst antibodies for antibodies that are randomly generated. Typical applications are described in [8, 23, 28], among others.

Those models have been used in several types of problems, but particularly, the use of artificial immune systems to solve constrained (numerical) optimization problems is scarce. The only previous related work that we found in the specialized literature is the following:

Hajela and Yoo [29, 30] have proposed a hybrid between a Genetic Algorithm (GA) and an AIS for solving constrained optimization problems. This approach works on two populations. The first is composed by the antigens (which are the best solutions), and the other by the antibodies (which are the worst solutions). The idea is to have a GA embedded into another GA. The outer GA performs the optimization of the original (constrained) problem. The second GA uses as its fitness function a Hamming distance so that the antibodies are evolved to become very similar to the antigens, without becoming identical. An interesting aspect of this work was that the infeasible individuals would normally become feasible as a consequence of the evolutionary process performed. This approach was tested with some structural optimization problems.

Kelsey and Timmis [18] proposed an immune inspired algorithm based on the clonal selection theory to solve multimodal optimization problems. Its highlight is the mutation operator called *Somatic Contiguous Hypermutation*, where mutation is applied on a subset of contiguous bits. The length and beginning of this subset is determined randomly.

Coello Coello and Cruz-Cortés [5] have proposed an extension of Hajela and Yoo's algorithm. In this proposal, no penalty function is needed, and some extra mechanisms are defined to allow the approach to work in cases in which there are no feasible solutions in the initial population. Additionally, the authors proposed a parallel version of the algorithm and validated it using some standard test functions reported in the specialized literature.

Balicki [2] made a proposal very similar to the approach of Coello Coello and Cruz-Cortés. Its main difference is the way in which the antibodies' fitness is computed. In this case, Balicki introduces a ranking procedure. This approach was validated using a constrained three-objective optimization problem.

Luh and Chueh [9, 21] have proposed an algorithm (called CMOIA, or Constrained Multi Objective Immune Algorithm) for solving constrained multiobjective optimization problems. In this case, the antibodies are the potential solutions to the problem, whereas antigens are the objective functions. CMOIA transforms the constrained problem into an unconstrained one by associating an interleukine (IL) value with all the constraints violated. IL is a function of both the number of constraints violated and the total magnitude of this constraint violation. Then, feasible individuals are rewarded and infeasible individuals are penalized. Other features of the approach were based on the clonal selection theory and other immunological mechanisms. CMOIA was evaluated using six test functions and two

structural optimization problems.

Coello Coello and Cruz-Cortés [6] have proposed an algorithm based on the clonal selection theory for solving constrained optimization problems. The authors experimented with both binary and real-value representation, considering Gaussian-distributed and Cauchy-distributed mutations. Furthermore, they proposed a controlled and uniform mutation operator. This approach was tested with a set of 13 test functions taken from the specialized literature on evolutionary constrained optimization.

In Section 4.5 we are going to explain the differences between the main models of AIS and our proposed model.

4 OUR PROPOSED MODEL

This paper presents a novel bio-inspired model based on the T-Cell, it is called T-Cell Model. In a very simple way, the processes that suffer the T-Cell are the following: first, they are divided in three groups (Virgin Cell, Effector Cells and Memory Cells). Then, the natural immune system generates a huge number of virgin cells. During the immunological response, the T-cells pass through different phases: initiation, reaction and elimination. After the initiation phase, virgin cells becomes effector cells. These react and undergo a process called $apoptosis$. This process eliminates any undesirable cells. The surviving cells become memory cells.

Thus, this model operates on three populations, corresponding to the three groups in which the T-cells are divided: (1) Virgin Cells (VC), (2) Effector Cells (EC) and (3) Memory Cells (MC). Each of them has a specified function. VC has as its main goal to provide diversity. EC tries to explore the conflicting zones of the search space. MC has to explore the neighborhood of the best solutions found so far. VC and EC represent their cells with binary string using Gray coding, MC does the same, but adopting vectors of real values. The general structure of this model is the following:

Repeat a predetermined number of times
1. Generate (in a random way) Virgin Cells
2. Insert a percentage of Virgin Cells in Effector Cells
3. Repeat a predetermined number of times
   3.1. Make the Effector Cells React
   End repeat.
4. Insert a percentage of Effectors Cells in Memory Cells
5. Repeat a predetermined number of times
   5.1. Make the Memory Cells React
   End repeat.

End repeat.

4.1 Handling Constraints

In our proposed model, the constraint-handling method needs to calculate, for each cell (solution) regardless of the population to which it belongs, the following: 1) value of each constraint function, 2) sum of violation constraints (sum_res), it is a positive value determined by the add of $g_i(x)^+$ for $i = 1, \ldots, p$ and $|h_k(x)|$ for $k = 1, \ldots, l$ and 3) value of objective function (only if the cell is feasible).

When the search process is driven by the value of each constraints and the sum of constraints violation, then the selection mechanisms favors the feasible solutions over the infeasible ones. In this case, it is probably that, in some functions, the search falls into a local optimum. For this reason, we develop a dynamic tolerance factor (DTF). It changes with each new population, since it depends on

the value of sum_res. The DTF is calculated by adding the value of each constraint violated in each cell from a particular population (VC or EC). Then, this value is divided by the number of Virgin Cells (for DTF's VC) or three times the number of Effector Cells (for DTF's EC).

When we evalue the population using the DTF, it will be easier to generate solutions that are considered "feasible" (although they may be really infeasible if evaluated with the actual precision required). This allows the exploration of each solution's neighborhood, which otherwise, would not be possible. This DTF is used by both VC and EC. If the value of DTF is lower than 0.0001, we set it to 0.1 and 0.001 for VC and EC, respectively. In constrast, MC adopts a traditional tolerance factor, which is set to 0.0001.

4.2 Incorporating Domain Knowledge

In order to explore the frontier between feasible and infeasible zones, EC is divided in EC_f and EC_inf. The first is composed by feasible solutions and the other by infeasible solutions. Also, we introduce domain knowledge through the mutation operator, which modify the decision variables involve in the constraint with the highest violationed.

4.3 Mutation Operators

Each population that reacts (EC_f, EC_inf and MC) has its own mutation operator. These operators are described next.

The mutation operator for EC_inf works in the following way: first, it identifies the most violated constraint, say c. If this constraint value (c) is larger than sum_res divided the total number of constraints, then we change each bit from each decision variable involve in c with a random probability between 0.01 and 0.2. Otherwise, we change each bit from one decision variable involve in c, randomly selected, with a random probability between 0.01 and 0.2. We use a random probability because after some experiments, we observed that some test functions required different step sizes. If after applying mutation, a cell becomes feasible, it is inserted in EC_f according to an elitist selection.

The mutation operator for EC_f works in the following way: it changes each bit from all decision variables, with a random probability between 0.001 and 0.2. This random probability has the same motivation that the previously.

The mutation operator for MC applies the following equation:

$$x^{'} = x \pm \left(\frac{N(0,1)lu - ll}{10^{m} gen|const||dv|} \right)^{N(0,1)} \quad (1)$$

where x and $x^{'}$ are the original and mutated decision variables, respectively. $N(0,1)$ refers to a random number with a uniform distribution between (0,1). lu and ll are the upper and lower limits of x. $|const|$ refers to the number of constraints. $|dv|$ refers to the number of decision variables of the problem, gen is the current generation number and m is an integer.

4.4 Replace Mechanisms

The replace mechanisms are always applied in an elitist way, both within a population and between different populations. They take into account the value of objective function or the sum of constraint violation, depending on whether the cell is feasible or infeasible, respectively. Additionally, we always consider a feasible cell as better than an infeasible one. Note that before a cell is inserted into another population, it is first evaluated with the tolerance factor of the receptor population.

Therefore, the general structure of our proposed model for constrained problems is the following:

Repeat a predetermined number of times
1. Randomly generate Virgin Cells
2. Calculate DTF's VC
3. Evaluate VC with its own DTF
4. Insert a percentage of Virgin Cells into Effector Cells population
5. Calculate DTF's EC's
6. Repeat 50 times
   6.1. Make the Effector Cells React
   6.2. Evaluate EC's with its own DTF
   End repeat.
7. Insert a percentage of Effectors Cells into Memory Cells population
8. Repeat 100 times
   8.1. Make the Memory Cells React
   8.2. Evaluate MC
   End repeat.

End repeat.

The most relevant aspects of our proposed model are the following:1) all equality constraints are converted into inequality constraints, $|h(\vec{x})| - \delta \leq 0$, using a tolerance factor, 2) VC's cells and MC's cell are sorted using the following criterion: the feasible cell whose objective function are the best are placed first. Then, we place the infeasible cells that have the lowest sum of constraint violation, 3) EC_f's cells are sorted in ascending order on their objective function y 4) EC_inf's cells are sorted in ascending order on their sum of constraint violation.

4.5 Differences between the Models

The immune system models described in 3 are based on different immunological theories. Clonal Selection is based on the replication of antibodies according to their affinity. The Immune Network Model is a probabilistic approach to idiotypic networks. Negative Selection is based on the principles of self-nonself discrimination that take place in the immune system. Aditionally, Negative Selection and T-Cell Model are both based on the mechanisms of the T-Cell. However, these models give a completely different treatment to the cells (in T-Cell Model) and detectors (in Negative Selection). The Negative Selection Model tries to detect some change, whereas T-Cell Models categorizes the T-cell and it uses their phases in order to achieve different goals.

5 EXPERIMENTAL SETUP

In order to validate our proposed model we tested it with a benchmark of 19 test functions taken from the specialized literature [4]. The functions g02, g03, g08 and g12 are maximization problems (for simplicity, these problems were converted into minimization problems using $-f(x)$) and the rest are minimization problems.

Our results are compared with respect to Stochastic Ranking, we take its result from [20], which is a constraint handling technique representative of the state-of-the-art in the area, and with respect to the AIS approach reported in [6]. For equation 1, we used $m = 10^7$ for all functions except for g02, here we used $m = 10^2$. 25 independents runs were performed for each problem, each consisting of 350,000 fitness function evaluations. We experimented with different population sizes, the best

results were obtained using: 1) for VC 100 cells for all functions, except for g19 here we used 10 cells and for g10 and g15 we used 20 cells, 2) for EC_f, EC_inf and MC we used 20 cells for all functions, except for g10 and g19, here we used 10 cells. We adopted a 100% and 50% replacement for the cells in EC's and MC, respectively. All the statistical measures reported are taken only with respect to the runs in which a feasible solution was reached at the end.

6 DISCUSSION OF RESULTS

Tables 1, 2 and 3 show the results obtained with the AIS proposed in [6], Stochastic Ranking and our T-Cell Model, respectively.

From Table 3, we can see that our model was able to reach the global optimum in 8 test functions (g01, g04, g06, g08, g11, g12, g15 and g16). Additionally, our model reached feasible solutions close to the global optimum in 7 more test functions (g02, g03, g07, g09, g13, g14 and g18) and it found acceptable (but not too close from the global optimum) feasible solutions for the rest of the test functions.

Comparing T-Cell Model with respect to Stochastic Ranking (see Tables 2 and 3), T-Cell Model obtained better results in 9 test functions (g03, g04, g06, g11, g14, g15, g16, g17 and g18). Both approaches found similar solutions for g01, g08 and g12. Our model was outperformed in 7 functions (g02, g05, g07, g09, g10, g13 and g19) . With respect to the mean and worst found solutions, our model was outperformed all functions except g03, g04, g06, g11, g14 and g16.

Comparing T-Cell Model with the AIS proposed in [6] (see Tables 1 and 3), T-Cell Model obtained better results in 8 test functions (g01,g02, g03, g05, g06, g07, g10 and g11). Both approaches found similar solutions for g04, g08 and g12. Finally, our model was outperformed in g09 and g13. With respect to the mean and worst found solutions, our model was outperformed only in g02, g07, g09 and g13.

We conducted an analysis of variance of the results obtained by our T-Cell Model and of the results obtained by Stochastic Ranking [20](we did not performe this analysis with the results obtained by the AIS proposed in [6] because the values required by the tests (for this approach) were not available). Due to, for some functions, the results do not follow a normal distribution, we used the Kruskal Wallis test and then Turkey method. The first test indicates if the means between the results of the algorithms had significant differences and the second one indicates in which experimental conditions the means had significant differences. Table 4 shows the values obtained for these tests. The first column represents to the function, the second column shows the values for Kruskal Wallis test (the means had significant differences if this value p is lower than 0.05), the third and fourth column indicate the lower and upper limits (if the values contained inside this interval does not contain the zero then the means had significant differences). After the analysis of Table 4, we observed that for all function the means have significant differences except for g11. Note that we do not apply these tests to g01, g08 ang 12 because, for these functions both algorithms found the optimum solution in all runs.

We argue that the model is capable of performing an efficient local search over each cell, which allows the model to improve on the feasible solutions found. In cases in which no feasible solutions are found in the initial population, the mutation applied is capable of reaching the feasible region even when dealing with very small feasible search spaces.

Although there is clearly room for improving our proposed model, we have empirically shown that this approach is able of dealing with a variety of constrained optimization problems (i.e., with both linear and nonlinear constraints and objective function, and with both equality and inequality constraints). The benchmark adopted includes test functions with both small and large feasible regions,

as well as a disjoint feasible region.

Table 1: Results obtained with AIS proposed in [6]. The asterisk (*) indicates a case in which only 90% of the runs converged to a feasible solution

Function	Optimum	*Best*	*Mean*	*Worst*	*Std.Dev*
g01	-15	-14.9874	-14.7264	-12.9171	0.6070
g02	-0.803619	-0.8017	-0.7434	-0.6268	0.0414
g03	-1.0005	-1.0	-1.0	-1.0	0.0000
g04	-30665.5386	-30665.5387	-30665.5386	-30665.5386	0.0000
g05*	5126.4967	5126.9990	5436.1278	6111.1714	300.8854
g06	-6961.81387	-6961.8105	-6961.8065	-6961.7981	0.0027
g07	24.306	24.5059	25.4167	26.4223	0.4637
g08	-0.095825	-0.095825	-0.095825	-0.095825	0.0000
g09	680.63	680.6309	680.6521	680.6965	0.0176
g10	7049.24	7127.9502	8453.7902	12155.1358	1231.3762
g11	0.7499	0.75	0.75	0.75	0.0000
g12	-1.0	-1.0	-1.0	-1.0	0.0000
g13	0.05395	0.05466	0.45782	1.49449	0.3790

Table 2: Results obtained with Stochastic Ranking [20]

Function	Optimum	*Best*	*Mean*	*Worst*
g01	-15	-15.0	-15.0	-15.0
g02	-0.803619	-0.803	-0.784	-0.734
g03	-1.0005	-1.0	-1.0	-1.0
g04	-30665.539	-30665.539	-30665.480	-30664.216
g05	5126.4967	5126.497	5130.752	5153.757
g06	-6961.81387	-6961.814	-6863.645	-6267.787
g07	24.306	24.310	24.417	24.830
g08	-0.095825	-0.095825	-0.095825	-0.095825
g09	680.63	680.63	680.646	680.697
g10	7049.24	7050.194	7423.434	8867.844
g11	0.7499	0.750	0.750	0.751
g12	-1.0	-1.0	-1.0	-1.0
g13	0.05395	0.053	0.061	0.128
g14	-47.7648	-41.551	-41.551	-40.125
g15	961.71502	961.715	961.731	962.008
g16	-1.905155	-1.905	-1.703	-1.587
g17	8853.539	8811.692	8805.99	8559.613
g18	-0.86602	-0.866	-0.786	-0.457
g19	32.655	33.147	34.337	37.477

Table 3: Results obtained with our proposed T-Cell Model. The asterisk (*) indicates a case in which only 96% of the runs converged to a feasible solution

Function	Optimum	*Best*	*Worst*	*Mean*	*Std.Dev*
g01	-15.0	-15.0	-15.0	-15.0	0.0
g02	-0.803619	-0.802914	-0.301795	-0.546031	0.168392
g03	-1.0005	-1.000499	-1.000498	-1.000499	0.000001
g04	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0
g05*	5126.4967	5126.6595	5850.9358	5307.1073	230.2466
g06	-6961.81387	-6961.81387	-6961.81387	-6961.81387	0.0
g07	24.306	24.3118	28.5089	25.8927	1.1297
g08	-0.095825	-0.095825	-0.095825	-0.095825	0.0
g09	680.63	680.6312	680.7411	680.6730	0.030547
g10	7049.24	7061.67	7894.75	7451.88	218.39739
g11	0.7499	0.7499	0.7499	0.7499	0.0
g12	-1.0	-1.0	-1.0	-1.0	0.0
g13	0.05395	0.054879	2.03033	0.64231	0.534641
g14	-47.7648	-46.2546	-40.2996	-43.6876	1.538386
g15	961.71502	961.71502	971.43611	065.02171	3.10270
g16	-1.905155	-1.905155	-1.905155	-1.905155	0.0
g17	8853.539	8862.383	9271.390	8984.399	117.5927
g18	-0.86602	-0.866019	-0.66920	-0.78805	0.09285
g19	32.655	34.649	73.151	52.617	10.1005

Table 4: Analysis of Variance

Function	p	lower limit	upper limit
g02	2.54392e-009	16.0252	31.7348
g03	4.53296e-011	-35.0356	-18.9644
g04	8.98673e-011	17.4421	32.5579
g05	2.17934e-009	16.2556	32.0911
g06	9.06124e-011	17.4406	32.5594
g07	2.93747e-009	15.4582	30.7018
g09	1.08889e-008	14.9037	30.4563
g10	1.74435e-008	14.5837	30.1363
g11	0.1298	-1.7753	13.8553
g13	3.35698e-010	17.1443	32.6957
g14	0.0009	-21.3391	-5.4609
g15	2.01142e-008	14.9660	31.0340
g16	3.97653e-011	-32.4185	-17.5815
g17	3.60989e-010	17.1853	32.8147
g18	7.25903e-010	16.6646	32.2157
g19	3.15542e-010	16.8670	32.1330

7 CONCLUSIONS AND FUTURE WORK

This paper has presented a new AIS model for solving constrained optimization problems in which novel mutation operators are adopted. One of the operators incorporates knowledge of the problem, by modifying the decision variables involve in the most violated constraint. For some functions, the feasible region is very small, wich makes it difficult to find good solutions. For this reason, we were motivated to develop a dynamic tolerance factor. It allows to explore regions of the search space that, otherwise, would be unreachable, if we use a tolerance factor very restrictive.

The proposed model was found to be competitive in a well-known benchmark commonly adopted in the specialized literature on constrained evolutionary optimization. The approach was also found to be robust and able to converge to feasible solutions in most cases.

Our analysis of the benchmark adopted made us realize that these test functions require small step sizes for mutation operators, except for g02, due to this function has a feasible region bigger than the other functions. A lot of work remains to be done in order to improve the quality of some solutions found, so that the approach can be competitive with respect to the algorithms representative of the state-of-the-art in the area. For example, we plan to improve the mutation operators in order to find the frontier and feasible zone faster. Nevertheless, it is important to emphasize that there is very little work regarding the use of artificial immune systems for constrained numerical optimization, and in that context, this approach provides a viable alternative.

REFERENCES

[1] Y. Watanable A. Ishiguru and Y. Uchikawa. Fault diagnosis of plant system using immune network. In Procceding of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent System (MFI'94), Las Vegas, October 2-5, 1994.

[2] Jerzy Balicki. Multi-criterion Evolutionary Algorithm with Model of the Immune System to Handle Constraints for Task Assignments. In Leszek Rutkowski, Jörg H. Siekmann, Ryszard Tadeusiewicz, and Lotfi A. Zadeh, editors, *Artificial Intelligence and Soft Computing - ICAISC 2004, 7th International Conference. Proceedings*, pages 394–399, Zakopane, Poland, June 2004. Springer. Lecture Notes in Computer Science. Volume 3070.

[3] Carlos A. Coello Coello. Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002.

[4] J. Liang T. Runarsson E. Mezura-Montes M. Clere P. Suganthan C. Coello Coello and K. Deb. Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. Technical Report, Nanyang Technological University, 2006.

[5] Carlos A. Coello Coello and Nareli Cruz-Cortés. Hybridizing a genetic algorithm with an artificial immune system for global optimization. *Engineering Optimization*, 36(5):607–634, October 2004.

[6] Nareli Cruz Cortés, Daniel Trejo-Pérez, and Carlos A. Coello Coello. Handling constrained in global optimization using artificial immune system. In Christian Jacob, Marcin L. Pilat, Peter J. Bentley, and Jonathan Timmis, editors, *Artificial Immune Systems. 4th International Conference, ICARIS 2005*, pages 234–247. Springer. Lecture Notes in Computer Science Vol. 3627, Banff, Canada, August 2005.

[7] D. Dasgupta and S. Forrest. An anomaly detection algorithm inspired by the immune system. In Artificial Immune System and Their Applications, chapter 14, pages 262-277. Springer-Verlag, Inc, 1999.

[8] L. de Castro and J. Timmis. Ainet: An artificial immune network data analysis. In Abbass, H. A., Sarker, R. A., and Newton, C. S., editors, Data Mining: A Heuristic Approach, chapter XII, pages 231-259. Idea Group Publishing, USA, 2001.

[9] H. Chueh G. C. Luh and W. W. Liu. MOIA: Multi Objective Immune Algorithm. *Engineering Optimization*, 35(2):143–164, 2003.

[10] Simon M. Garrett. How do we evaluate artificial immune systems? *Evol. Comput.*, 13(2):145–177, 2005.

[11] S. Hofmeyr and S. Forrest. Architecture for the artificial immune system. Evolutionary Computation, 8(4):443-473,2000.

[12] J. E. Hunt and D. E. Cooke. An adaptative, distributed learning system based on the immune system. In Procceding of the IEEE International Conference on System, Man and Cybernatics, pages 2494-2499, 1995.

[13] J. E. Hunt and D. E. Cooke. Recognising promoter sequences using immune algorithms. In Procceding of the 3rd IEEE International Conference on Intelligent System for Molecular Biology (ISMB), pages 89-97, 1995.

[14] J. E. Hunt and A. Fellows. Introducing an immune reponse into a cbr system for data mining. 1996.

[15] Y. Ishida. An immune network model and its applications to process diagnosis. System and Computer in Japan, 24(6):646-651, 1993.

[16] M. Neal J. Timmis and J. Hunt. An artificial immune system for data analysis. Biosystem, 55(1/3):143-150, 2000.

[17] N. K. Jerne. The imune system. Scientific American, 229(1):52-60, 1973.

[18] Johnny Kelsey and Jon Timmis. Immune inspired somatic contiguous hypermutation for function optimisation. In Erick Cantú-Paz, James A. Foster, Kalyanmoy Deb, Lawrence Davis, Rajkumar Roy, Una-May O'Reilly, Hans-Georg Beyer, Russell K. Standish, Graham Kendall, Stewart W. Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, Mitchell A. Potter, Alan C. Schultz, Kathryn A. Dowsland, Natasa Jonoska, and Julian F. Miller, editors, *GECCO*, volume 2723 of *Lecture Notes in Computer Science*, pages 207–218. Springer, 2003.

[19] Slawomir Koziel and Zbigniew Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7(1):19–44, 1999.

[20] S. Esquivel L. Cagnina and C. Coello Coello. A bi-population pso with a shake-mechanism for solving numerical optimization. In Proceedings of the Congress on Evolutionary Computation 2007. Accepted. To appear.

[21] G. C. Luh and H. Chueh. Multi-objective optimal designof truss structure with immune algorithm. *Computers and Structures*, 82:829–844, 2004.

[22] L. Nunes de Castro and J. Timmis. An artificial immune network for multimodal function optimization. In *Proccedings of the 2002 Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 669–674, Honolulu, Hawaii, May 2002.

[23] L. Nunes de Castro and F.J. Von Zuben. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251, 2002.

[24] Thomas P. Runarsson and Xin Yao. Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.

[25] L. Allen S. Forrest, A. Perelson and R. Cherukuri. Self-nonself discrimination in a computer. *IEEE Symposium on Research in Security and Privacy*, pages 202–212, May 1994.

[26] Alice E. Smith and David W. Coit. Constraint Handling Techniques—Penalty Functions. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter C 5.2. Oxford University Press and Institute of Physics Publishing, 1997.

[27] J. Timmis and M. Neal. A resource limited artificial immune system for data analysis. In Procceding of ES2000, pages 19-32, Cambridge, UK, 2000.

[28] J. White and S. Garret. Improved pattern recognition with artificial clonal selection. In Proceding of the 2nd International Conference on Artificial Immune Systems (ICARIS-03), pages 181-193, 2003.

[29] J. Yoo and P. Hajela. Enhanced GA Based Search Through Immune System Modeling. In *3rd World Congress on Structural and Multidisciplinary Optimization*, Niagara Falls, New York, May 1999.

[30] J. Yoo and P. Hajela. Immune network modelling in design optimization. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 167–183. McGraw-Hill, London, 1999.

Knowledge Dynamics in a Khepera Robots' Application

Luciano Héctor Tamargo **Alejandro Javier García** **Marcelo Alejandro Falappa**
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Laboratorio de Investigación y Desarrollo de Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur
Avenida Alem 1253,(B8000BCP), Bahía Blanca, Argentina
Tel: (0291) 459-5135 / Fax: (0291) 459-5136
{lt,ajg, mfalappa}@cs.uns.edu.ar

Abstract

The theory change area have defined several operators trying to model the dynamic of knowledge. The goal of this work is to introduce a *preprocessor of perceptions* that could be part of an agent. The purpose of the preprocessor is to select an adequate change operator to store the perceptions that the agent perceives from the environment where their activities are performed. An agent receives perceptions, and classifies them according to how they were obtained by him/her. Once are the perceptions classified, the preprocessor selects one of the change operators to store the new information in the belief base. Then the *preprocessor of perceptions* sends the selected method and the perceptions to a change machine in the agent. When the change machine receives the mentioned inputs, it interacts with the belief base of the agent to store the new information.
Key words: Multi-Agent Systems, Khepera, Preprocessor, Revision, Update.

1 INTRODUCTION

In a multi-agent system with a dynamic environment, agents should store the information that they perceive from the environment where they perform their activities. In order to achieve this, the agents should have a change machine, that will interact with the belief base of the agent, and some storing mechanism to store the new information that perceive from the environment. Furthermore, this mechanism should conserve the consistency of the belief base. Therefore, we will base our work upon the theory change. The main goal of this theory is to model the dynamic of knowledge. That is, how the agents beliefs are defined after they receive the out-source information.

Khepera robots [1] have multiple sources to obtain knowledge from the environment. They have eight infrared sensors and a video camera (for a more detail reading, see Section 4). They use these mechanisms to perceive information from the environment in which they perform their activities. In this work, we will show the proposal through examples including Khepera robots. These examples consider a simple situation of the *"cleaning task problem"* where one robot has to transport to a particular place (store), a particular box.

In theory change have been proposed different change models. The AGM model [2] is one of their main referents. This model belongs to the theory revision, and it distinguishes three change operators: *expansions*, *contractions* and *revisions*. Katsuno and Mendelzon in [11] distinguish other

Partially supported by CONICET (PIP 5050), Universidad Nacional del Sur and Agencia Nacional de Promoción Científica y Tecnológica.

type of operators. These are called *updating* and *erasure* (they belong to the theory updating). They model those changes in the world. Furthermore, as we will show below, in the theory revision there are defined several operators with different semantics.

The goal of this work is to introduce a *preprocessor of perceptions* that could be part of an agent. The purpose of this preprocessor is to select an adequate change operator to store the perceptions that the agents perceive from environment where their activities are performed. This preprocessor could be characterized by a framework which is defined as: $< \mathcal{L}, Op, COA >$ where $\mathcal{L}$ is a generic language (the language will not be fully considered in this work), Op is a set of change operators, and COA is a credibility order among agents.

The *preprocessor of perceptions* receives Op, COA and perceptions (sentences belong to the language $\mathcal{L}$), and classifies the perceptions according to how they were obtained by the agent. Once the perceptions are classified, the preprocessor selects one of the change operators from Op, to store the new information in the belief base. If the perception comes from an agent, this selection could depend of a credibility order among agent defined in COA. Then the preprocessor sends the selected method and the perceptions to a change machine in the agent. When the change machine receives the mentioned inputs, interacts with the belief base of the agent to store the new information. In Figure 1 a general scheme of the proposal is depicted.

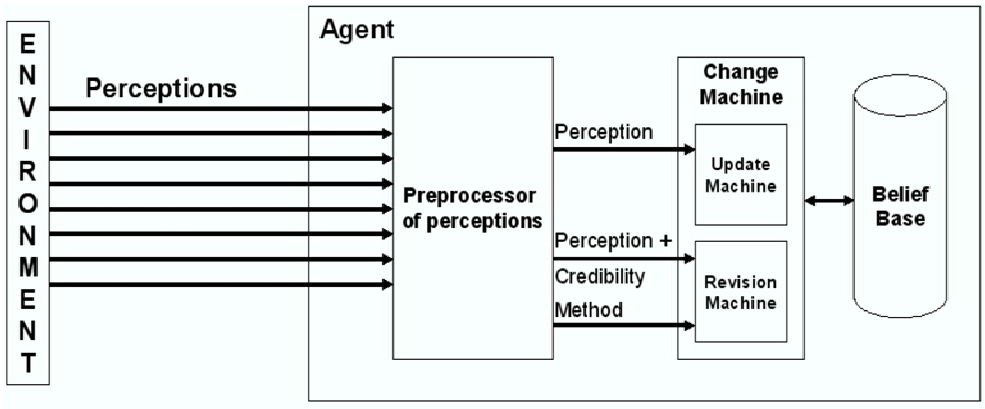


Figure 1: General scheme of the proposal

As shown in Figure 1, the change machine is splitted in two parts, update machine and revision machine. Both machines interact with the belief base to consistently store perceptions, either through an update operator or a revision operator, respectively. The update machine only receives one input, the perception. The revision machine receives two inputs, the perception and its credibility (below we will see that the sentences will have credibility too), and a method. This is because in the theory revision exists several change operators.

The work is organized as follows: to next, we will describe a general example to motive the proposal. In Section 2, a background of theory revision will be given, describing notions about epistemic model, updating and revision. In Section 3, we will introduce the *preprocessor of perceptions* characterized by a multi-change framework. In Section 4, a brief description over Khepera robots will be given, and we will show how the behavior of the preprocessor in the Khepera should be. In Section 5 we will give the conclusions about this work.

Example 1 *Lets suppose we have a multi-agent system with a dynamic environment. The system is composed by Khepera 2 robots [1] and the experimental environment is a square arena. The Khepera robots have infrared sensors and a video camera (for more detail see Section 4). They use*

these mechanisms to perceive information from the environment where their activities are performed. In the arena there are boxes, obstacles and a particular place (store) where they should deposit a determined box. Khepera robots start in an initial position in the arena and they should search a particular box. Once the particular box is found, the robots will transport it towards the store. When the robots start their activities, their knowledge will be submitted to subsequent revision processes. That is, the Khepera incorporate beliefs such as: 'the store is in the position (x,y)', 'there is a box in the position (x,y)' or any related information represented in a determined language. The robot will acquire knowledge to find the particular box and to achieve its goal.

2 THEORY CHANGE

The main goal of theory change is to intent to model the dynamic of knowledge. That is, how the agents beliefs are defined after they receive the out-source information. In theory change have been proposed different change models. The AGM model [2] is one of their main referents. This model belongs to the theory revision, and it distinguishes three operations of change: *expansions*, *contractions* and *revisions*.

2.1 Epistemic Model

The **epistemic states** represent states of the knowledge of an agent in a moment of time. Generally, we expect that the epistemic state will be equilibrated. That is, we expect that the epistemic state will not have contradictory beliefs. An equilibrated epistemic state does not need to be evaluated because it does not have contradictory beliefs. The epistemic state is evaluated when the agent receives a **epistemic input** of external information that it contradicts the current epistemic state. In this case, we should modify the new epistemic state until lead it to an equilibrated state.

In an idealized interpretation of the epistemic state of an agent, we assume that this is in an equilibrated state. Changes of a belief state are produced by some epistemic input that force the change. It is desirable that the new belief state will be equilibrated too. Our interest is centered in the effect produced by epistemic inputs over the epistemic state. The operation that adds information to the epistemic state and then it modifies the epistemic state to lead it to a equilibrated state is called "*revision*" or "*update*" (this depends of the operator type that should be used).

In this work we consider that the epistemic states of agents are represented with **Belief Bases** and not with closed sets under logic consequents. The Belief Bases are represented through a set of sentences not necessarily closed under logic consequence. This feature makes the change operations computationally tractable over belief states (represented through belief bases). This could make us think that our effort in the development of change operations should be centered in belief bases and not in belief sets. However, we will see that this first impression seem hurried. The models that employ belief bases are based in the intuition of that some of our beliefs do not have a independent sustentation. That is to say, these belief types appear as a consequence of applying inference rules over our beliefs of which they totaly depend [4].

2.2 Revision

The *revision* is one of the most common operations that an agent develops in an dynamic environment where his/her activities are performed. Whether an agent believes in a set K, and the agent perceives an epistemic input α, he/she should revise the beliefs of K. The agent does this in order to determine

which of his/her beliefs are in disagreement with α. This is done eliminating some beliefs that contradict α and adding α afterwards. This process generates a new consistent K'. In this context, different situations may arrive:

- The epistemic input has primacy [3] over the epistemic state of the agent. In this case we should eliminate of the epistemic state all beliefs that are inconsistent with the input.
- The epistemic input does not have primacy. In this case, the agent may preserve his/her epistemic state (ideally equilibrated), because the agent could not consider that the epistemic input is more credibility than the epistemic state.

The second case motivated the development of other types of revision operators called "non prioritized". Among these, we may find the *revision by a set of sent* [5]. This is a revision operator non prioritized that demands that the external beliefs were supported in explanations. The acceptation or not of the external belief will depend of the explanation: whether the explanation is sufficiently convincing, the new belief (and its justification) will be accepted; else it will not have changes in the knowledge [5].

When an agent perceives a new epistemic input is important to distinguish two cases. Let suppose the epistemic state of an agent is represented through a set of sentences K. Suppose that α is the input sentence. The cases to consider in a epistemic change are the following:

- When α is consistent with K.
- When α contradicts K.

In the first case to maintain equilibrated the new epistemic state not seem something complicated. Only we should add α to the belief base. This is a operation of *expansion*. However, when α contradicts to K then the negation of α is inferred from K. Hence, before of to incorporate α to K, we should eliminate of K some or all the sentences that allow the deduction of $\neg\alpha$ in K. This change operation is called *contraction*. Furthermore, to complete the change in the epistemic state we should add the sentence (the epistemic input) to the contracted set. This type of operation is called *revision*.

2.3 Updating

The change operations that we have mentioned above allow to modify the agent knowledge. The goal of the operations is to dynamically model the knowledge of an agent when new information arises. For instance, consider the multi-agent system of Example 1. The robots will acquire knowledge to find the particular box. They will assume that the environment always will be the same. That is, in the environment (where the agents perform their activities) does not occur change.

Unfortunately, the previous assumption is very strong. The world, where an agent performs his/her activities, may change by its own evolution, by actions of the agent, or by actions of other agents. For this matter, the change operations that we have shown above (*expansion*, *contraction* and *revision*) allow to modify the knowledge of an agent assuming that the world is static. That is, there are changes in the knowledge but not in the world. Katsuno and Mendelzon distinguish other type of operators [11]. These are called *updating* and *erasure*. They model those changes in the world, and this involves that the environment in which the agents perform their activities is dynamic and may change by its own evolution or by actions of other agents. In the example of the Khepera robots, in the following case some changes in the world will occur: we assume that there are two Khepera robots (A and B) in the arena and that there are boxes that may be transported by the robots. In this

case, whether an agent A transports a box by his/her own action and an agent B passes for second time for the place where the box was, the agent B perceives one change in the world and will update his/her knowledge.

The operations to model changes in the world are called *updating* and *erasure*. They are similar to the operations of *revision* and *contraction* that allow to model changes in the knowledge of an agent. This is the reason from which some authors clearly distinguish the **theory revision** (called theory change too) of the **theory updating**. In the theory updating are developed *update* operators and from these are formulated the *erasure* operators. In counterpart, most of the belief revision systems define *contraction* operators in order to define *revision* operators. Other difference is that the models of revision are defined syntactically (except the Grove construction [7] and the work of Katsuno and Mendelzon [12]), and the update operations are defined semantically. Next, definitions of update and revision proposed by Katsuno and Mendelzon in [11] are shown:

- Update: Consists of bringing the knowledge base up to date when the world described by it changes. For example, the incorporation into the knowledge base of changes caused in the world by the actions of a robot. The AGM postulates must be drastically modified to describe update.

- Revision: is used when we are obtaining new information about a static world. For example, we may be trying to diagnose a faulty circuit and want to incorporate into the knowledge base the results of successive tests, where newer results may contradict old ones.

Previously we have analyzed some difference between revision operators and update operators. For the goal of this work, we consider that the most important difference is the next one: the type of change operators formulated from the theory revision, characterize the changes produced in the knowledge of an agent. However, they do not give a formal mechanism to model the changes in the world. This is modeled by theory updating. The AGM model allows to model the changes in the knowledge of an agent assuming that the world does not change.

3 PREPROCESSOR OF PERCEPTIONS

As stated above, in the theory change have been proposed several operators that allow to incorporate information. These operators have similar aspects (they allow to incorporate a belief and hold the belief base consistent), but their semantics differs. That is, each type of operator defined in the literature has distinctive features. These features determine which operator to select. Next we will define a *preprocessor of perceptions*. This preprocessor could be part of the agents and will allow them to decide which operator to use in a particular situation.

Definition 1 : **(Preprocessor of perceptions)** The *preprocessor of perceptions* is a selection procedure composed by two modules, a *classifier of perceptions* and an *analyzer of senders*. It receives a set of change methods, a credibility order among agents, perceptions and it returns the change operator to be applied to consistently store the perceptions.

The purpose of this preprocessor is to select an adequate change operator to store the perceptions that the agents perceive from environment. This preprocessor could be characterized by a framework which is defined as: $< \pounds, Op, COA >$ where $\pounds$ is a generic language, Op is a set of change operators, and COA is a credibility order among agents.

The *preprocessor of perceptions* receives Op, COA and perceptions (sentences belong to the language $\mathcal{L}$), and classifies the perceptions according to how they were obtained by the agent (via sensors or via interaction with others agents). This classification is performed by a *Classifier of Perceptions* that is a small part of the preprocessor (this is defined below). If the classifier has selected a operator of the theory revision, then the preprocessor should newly analyze the perception to determine that revision operator to apply. This analysis is performed by a *analyzer of senders* (described below) with help of a credibility order among agents defined in COA. As secondary effect, the analyzer adds credibility to the sentences that represent the perceptions.

Once the perceptions are classified, the preprocessor selects one of the change operators from Op, to store the new information in the belief base. Then the framework sends the selected method and the perceptions to a change machine in the agent. When the change machine receives the mentioned inputs, interacts with the belief base of the agent to store the new information. In Figure 2 a detailed scheme of the proposal is depicted.

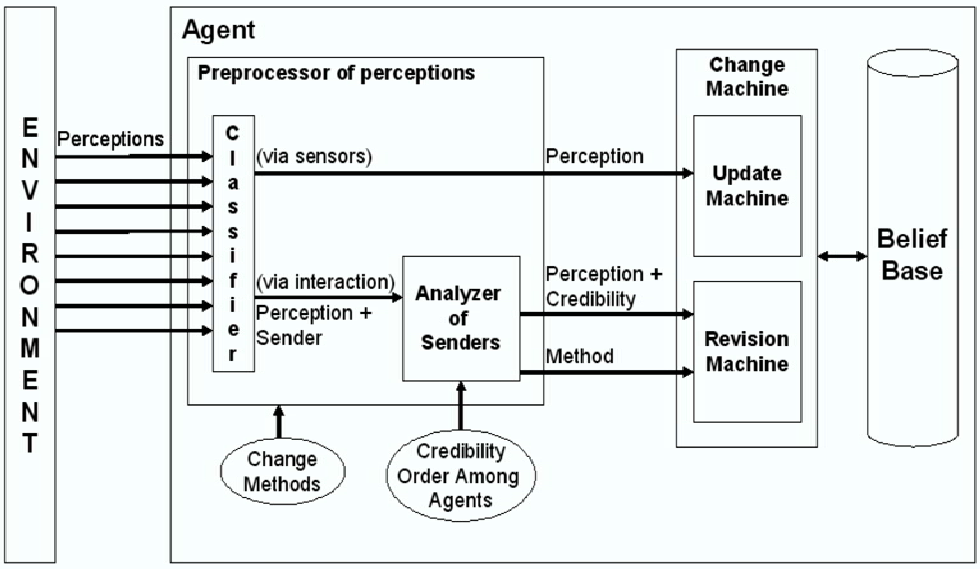


Figure 2: Detailed scheme of the proposal

As shown in Figure 2, the change machine is splitted in two parts, update machine and revision machine. Both machines interact with the belief base to consistently store perceptions, either trough an update operator or a revision operator, respectively. The update machine only receives one input, the perception. The revision machine receives two inputs, the perception and its credibility (below we will see that the sentences will have credibility too), and a method. This is because in the theory revision exists several change operators.

3.1 Classifier of Perceptions

Before determining which change operator should apply, the preprocessor should select the theory change to use. In order to achieve this, the preprocessor checks whether:

- the perceptions were obtained by sensors, (a mechanism to capture information from the environment such as infrared, video camera, etc.), or
- the perceptions were obtained by interaction with other agents.

In first case the perceptions will be classified as *"via sensors"*. Here, the preprocessor is going to consider that the operator most convenient is *updating*, because the agents detect the changes in the world through sensors. In second case the perceptions will be classified as *"via interaction"*. Here, the preprocessor is going to consider that the operator most convenient is some of the theory revision, because the perceptions are information that the agent received from other agent and maybe these perceptions do not represent some change in the world. In Figure 3 this is shown. In Definition 2 we define the module of the preprocessor that is responsible of this check.

Classification	**Selected theory by classifier**
via sensors	updating
via interaction	revision

Figure 3: How the classifier selects the adequate theory change

Definition 2 : **(Classifier of perceptions)** The *classifier of perceptions* is a small part of the *preprocessor of perceptions*. This receives the perceptions and performs a simple analysis to determine the theory change to be used, theory update or theory revision.

If the perception has been classified as *"via sensors"* then the preprocessor sends a message to the change machine (in this case, update machine) only with the perception. After this, the change machine interacts with the belief base of the agent to consistently store the new information. Otherwise, if the perception has been classified as *"via interaction"* then the preprocessor should newly analyze the perception to determine that revision operator should be applied. This is performed by a *analyzer of senders* (depicted below) with help of a credibility order among agents defined in COA. For this matter, the classifier sends to the analyzer two parameters, the perception and the identifier of its sender.

3.2 Analyzer of Senders

After the *classification of perceptions*, the preprocessor has already obtained which type operator to use, an operator of the theory revision or an operator of the theory updating. Whether revision operator has been selected, then the preprocessor should select which revision operator to apply. In order to achieve this, the preprocessor should perform a new analysis. This is performed by a module that we call *Analyzer of Senders*. This analyzer compares the credibility of the receiver agent with the credibility of the sender agent (the sender identifier is received from the *classifier of perceptions*). The comparison is based in a *credibility order among agents* defined in COA. Both, the analyzer and the order are defined to next.

Definition 3 : **(Analyzer of Senders)** The *Analyzer of Senders* is a part of the *preprocessor of perceptions*. This receives a credibility order among agents and perceptions (previously classified as *"via interaction"*), and return the revision operator to be applied based in the senders.

Definition 4 : **(Credibility Order among Agents)** Let $\mathcal{A}$ be the set of all the agents in a multi-agent system and Ag_1, Ag_2 and $Ag_3 \in \mathcal{A}$. A credibility order among agents '$\leq$' over $\mathcal{A}$ is a binary relation on $\mathcal{A}$ where $Ag_1 \leq Ag_2$ means that Ag_2 is more credible than Ag_1. Besides, $Ag_1 \leq Ag_2$ if and only if $Ag_1 \leq Ag_2$ and $Ag_2 \not\leq Ag_1$.

We assume that this order is total because for every Ag_1, $Ag_2 \in \mathcal{A}$, either $Ag_1 \leq Ag_2$ or $Ag_2 \leq Ag_1$. It satisfies the followings properties:

- Reflexive: $Ag_1 \leq Ag_1$.
- Transitive: if $Ag_1 \leq Ag_2$ and $Ag_2 \leq Ag_3$ then $Ag_1 \leq Ag_3$.
- Antisymmetry: if $Ag_1 \leq Ag_2$ and $Ag_2 \leq Ag_1$ then $Ag_1 = Ag_2$.

The mentioned analysis, that compares the credibility between receiver agent and sender agent, may return three results:

- The sender is more credible than receiver.
- The receiver is as credible as the sender.
- The receiver is more credible than sender.

Based in these results the analyzer may indicate which of type revision operator should be applied. In first case, the *analyzer of senders* will consider that the revision operators most indicated are those where the input has primacy over the belief base. For instance, *"Partial Meet Revision"* defined by AGM in [2], or *"Kernel Revision"* defined in [8]. In this work we consider that the selected operator by the analyzer should be *"Kernel Revision"* because it is more tratable over belief base.

In second and third cases, the *analyzer of senders* will consider that the revision operators most indicated are those where the input does not have primacy over the belief base. There are a lot of works related to non prioritized revision such as, *"Semi-Revision"* [9], *"Screened Revision"* [13], *"Credibility Limited Revision"* [10], *"Selective Revision"* [6] and *"Revision by a set of sentences"* [5]. In this work we consider that, in second case, the selected operator by the analyzer should be *"Selective Revision"*. Because if both agents, the receiver and the sender, have the same credibility level then the receiver should incorporate only part of the input. Besides we consider that, in third case, the analyzer should select any of the others non prioritized operators. Next in Figure 4 a brief of this will be shown.

Credibility Order	**Operators most convenient selected by analyzer**
The sender is more credible than receiver	Kernel Revision
The sender is as credible as receiver	Selective Revision
The receiver is more credible than sender	Semi-Revision, Screened Revision, Credibility Limited Revision, Revision by a Set of Sentences

Figure 4: How the analyzer selects the adequate revision operator

Observation 1: If the method to be used is the *"Revision by a Set of Sentences"* then the epistemic input should be supported by an explanation.

Once the preprocessor has analyzed the credibility among agents, the analyzer adds credibility to the perceptions. This credibility depends of the sender and the credibility order among agents. Thus, the perceptions will be pairs of the form (perception, sender identifier). This will allow that the change machine may compare the sentences through the credibility order and the identifier sender.

Once the credibilities of sentences have been added, then the *analyzer of senders* sends the perceptions, their credibilities and the selected method to the change machine (in this case, revision machine). The change machine interacts with the belief base of the agent to consistently store the new information through the method received.

Credibility of perceptions may be used by the change machine in the process of revision. That is, each method of the change machine will receive a perception and its credibility and each method will use the credibility in a different way. The credibility of the sentences may change when the contents of the belief base are modified.

Observation 2: The perceptions received by the update machine will have the greatest credibility because they determine change in the world (these sentences are obtained by agent *"via sensors"*). Hence the update machine does not need to receive their credibilities.

4 KHEPERA 2 ROBOT OVERVIEW

The *Khepera 2* robot [1], is a miniature mobile robot that allows confrontation to the real world of algorithms developed in simulation for trajectory execution, obstacle avoidance, pre-processing of sensory information, hypothesis on behaviors processing, among others. Its small size (60 mm diameter, 30 mm height), light weight (approx. 70 grams), and compact shape are ideal for micro-world experimentation. The *Khepera 2* has eight infrared sensors to sense both ambient light levels and proximity to nearby objects. It also has two DC motors that are capable of independent variable speed motion, allowing the robot to move forward, backward, and complete a variety of turns at different speeds.

The *Khepera 2* has several extension modules that can be plugged into the top of the robot. These include an arm with a gripper, a linear vision system, and a matrix vision camera. The *Khepera 2* has an on-board Motorola 68331 (25MHz) processor, 512 KB RAM, 512 KB Flash memory programmable via serial port, and rechargeable NiMH batteries that allows it up to 60 minutes of autonomy. Thus, the *Khepera 2* has sufficient sensors and actuators to ensure that it can be programmed to complete a wide variety of tasks.

4.1 The preprocessor in the Khepera robots

As stated above the Khepera robot has infrared sensors and a video camera. Besides, these robots may interact among them. These are three fashions that the Khepera robots has to perceive information from the environment. Hence, the problem that we intent to solve in this work, in the context of Khepera, consists in the following question: How does the epistemic state of an Khepera change when he/she perceives an epistemic input?.

In other words, based in the theory change, depending of the type of perception that the Khepera perceive, the robot should choose some change operator to consistently store the new information. These perceptions will be stored in the belief base thus altering the epistemic state of the agent. In order to achieve this, each Khepera could possess one *"Preprocessor of Perceptions"*.

Initially the perceptions (or epistemic input) will be classified by the preprocessor of the Khepera according to how they were obtained by the agent (*i.e.* robot). Hence, in case of the Khepera, a robot may receive perceptions of three difference ways:

- From infrared: these perceptions are those that were obtained by the proximity infrared sensors of the Khepera.

- From video camera: these perceptions are those that were obtained by the video camera of the Khepera.
- From interaction: these perceptions are those that were obtained by interaction with others agents.

In first and second case, the *classifier of perceptions* of the preprocessor classifies these perceptions as *"via sensors"*. In third case, the classifier classifies these perceptions as *"via interaction"*. If the input epistemic is classified as:

- *"via sensors"*, then we may tell that this input determines a change in the world. Hence the preprocessor of the Khepera will consider that the type operator most indicated should be some of theory updating.
- *"via interaction"*, then we may tell that this input determines a change in the agent beliefs but does not determine a change in the world. Hence the preprocessor of the Khepera will consider that the type operator most indicated should be some of theory revision.

Once this has been done, the preprocessor will perform its activities as we have detailed sections above. Next, we will present some examples that will describe the behavior of the preprocessor in the Khepera robots.

4.2 Examples

Lets suppose we have a multi-agent system with a dynamic environment as stated above in Example 1. Lets suppose in the arena there are 2 Khepera robots (Ag_1 and Ag_2). The goal of the Khepera is to find the particular box and transports it towards the store. In this context, we will depict two examples that they describe the behavior of the preprocessor in the Khepera robots. Each example describes different situations. In first example we will show the behavior of the preprocessor when an agent perceives information through interaction with other agent. In that case the new information is stored with a revision operator. In second example we will show the behavior of the preprocessor when an agent perceives information through his/her proximity infrared sensors. In this case the new information is stored with an update operator.

Example 2 *Lets suppose Ag_1 has the following belief in his/her base: 'the particular box is in the position (10, 20)'. Besides, lets suppose the agent Ag_1 starts interacting with the agent Ag_2. Through this interaction, Ag_1 perceives the following input: 'the particular box is in the position (10, 15)'. When this occurs, the preprocessor of Ag_1 receives the input perception and analyze it through the "classifier of perceptions". This classifies the perception as 'via interaction' because the input came from an interaction with other agent. Then the classifier considers that the most convenient type operator to be used should be any of the theory revision's. Hence, the classifier sends the perception and the sender identifier of Ag_2 to the "analyzer of senders". This takes the sender identifier and evaluates the credibility order between Ag_1 and Ag_2. In case that $Ag_1 \leq Ag_2$ (i.e. Ag_2 is more credible than Ag_1) the analyzer will consider that the operator to be used is "Kernel revision" [8]. In case $Ag_1 = Ag_2$ (i.e. Ag_1 is as credible as Ag_2) the analyzer will consider that the operator to be used is "Selective revision" [6]. In case that $Ag_2 \leq Ag_1$ (i.e. Ag_1 is more credible than Ag_2) the analyzer will consider that the operator to be used is "Revision by a set of sentences" [5]. After this, the preprocessor sends a message to the change machine (in this case, revision machine) with the perception and its credibility, and the selected method. Then, the change machine interacts with*

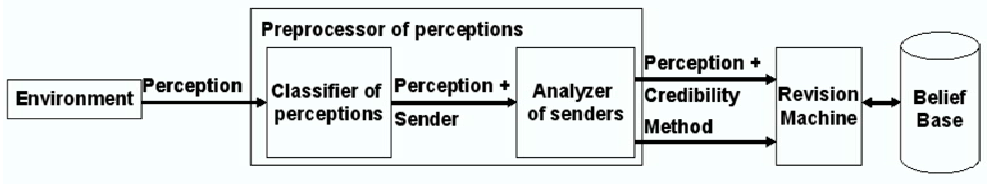


Figure 5: Passway of the perception through the preprocessor

the belief base of the agent to consistently store the new information. In Figure 5 the passway of the perception through the preprocessor is depicted.

Example 3 *Lets suppose Ag_1 has the following belief in his/her base: 'the particular box is in the position (10, 20)'. Besides, lets suppose the agent Ag_2 transports the particular box from the position (10, 20) to the position (15, 30). When Ag_1 newly finds the particular box, he/she will note a change in the world. In this situation the preprocessor of Ag_1 receives the input perception ('the particular box is in the position (15, 30)') and analyzes it through the "classifier of perceptions". This classifies the perception as 'via sensors' because the input was perceived through the proximity infrared sensors of the Khepera. Then the classifier considers that the most convenient type operator to be used should be any of the theory update's. Hence, the preprocessor sends a message to the change machine (in this case, update machine) only with the perception. After this, the change machine interacts with the belief base of the agent to consistently store the new information. In Figure 6 the passway of the perception through the preprocessor is depicted.*

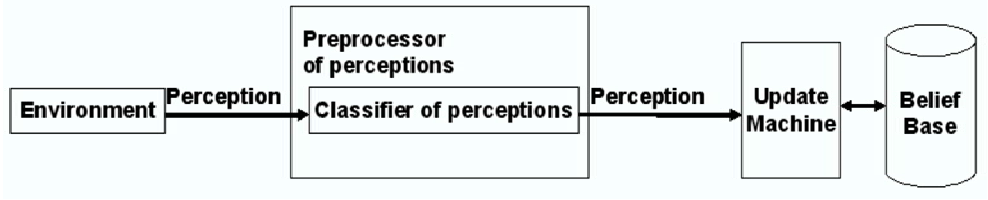


Figure 6: Passway of the perception through the preprocessor

5 CONCLUSION AND FUTURE WORK

In this work we have introduced a *preprocessor of perceptions* that could be part of an agent. The purpose of this preprocessor is to select an adequate change operator to consistently store the perceptions that the agents perceive from environment where their activities are performed. This preprocessor could be characterized by a framework which is defined as: $< \mathcal{L}, Op, COA >$ where $\mathcal{L}$ is a generic language, Op is a set of change operators, and COA is a credibility order among agents.

The *preprocessor of perceptions* receives Op, COA and perceptions (sentences that belong to the language $\mathcal{L}$), then classifies the perceptions according to how they were obtained by the agent (via sensors or via interaction with others agents). This classification is performed by a *Classifier of Perceptions* that is a small part of the preprocessor. If the classifier has selected a operator of the theory revision, then the preprocessor should newly analyze the perception to determine that revision operator should be applied. This analysis is performed by a *analyzer of senders* with help of a credibility order among agents defined in COA. As secondary effect, the analyzer adds credibility to

the sentences that represent the perceptions. This credibility depends of the sender and the credibility order among agents. The credibility of the sentences may change when the contents of the belief base are modified.

This preprocessor is very useful because it will allow to the agents to decide which operator to use in some determined case. That is, in different situations the preprocessor will use different change operators to consistently store the new information perceived from the environment. As future work, we plan to define properties over the framework that characterizes to the preprocessor and we think to fix $\mathfrak{L}$ in a more expressive language than a propositional one. Besides, we will focus our work in the credibility of the sentences and in the *Revision by a set of sentences*.

REFERENCES

[1] K-Team: Khepera 2. http://www.k-team.com. 2002.

[2] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

[3] Mukesh Dalal. Investigations into a theory of knowledge base revision. In *AAAI*, pages 475–479, 1988.

[4] Marcelo A. Falappa. Teoría de Cambio de Creencias y sus Aplicaciones sobre Estados de Conocimiento. *Tesis Doctoral desarrollada en el Grupo de Investigación en Inteligencia Artificial en la Universidad Nacional del Sur*, 1999.

[5] Marcelo A. Falappa, Gabriele Kern-Isberner, and Guillermo Ricardo Simari. Explanations, belief revision and defeasible reasoning. *Artificial Intelligent*, 141(1):1–28, 2002.

[6] Eduardo L. Fermé and Sven Ove Hansson. Selective revision. *Studia Logica*, 63(3):331–342, 1999.

[7] Adam Grove. Two modelings for theory change. *The Journal of Philosophical Logic*, 17:157–170, 1988.

[8] Sven Ove Hansson. Kernel contraction. *Journal of Symbolic Logic*, 59(3):845–859, 1994.

[9] Sven Ove Hansson. Semi-revision. *Journal of Applied Non-Classical Logic*, pages 151–175, 1997.

[10] Sven Ove Hansson, Eduardo L. Fermé, John Cantwell, and Marcelo A. Falappa. Credibility limited revision. *Journal of Symbolic Logic*, 66(4):1581–1596, 2001.

[11] Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In *KR*, pages 387–394, 1991.

[12] Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligent*, 52(3):263–294, 1992.

[13] David Makinson. Screened revision. *Theoria: Special Issue on Non-Prioritized Belief Revision*, 1997.

Continuous Evolution of Neural Modules for Autonomous Robot Controllers

Hernán Vinuesa[1], Germán Osella Massa[2], Leonardo Corbalán[3], Laura Lanzarini[4]
{hvinuesa, gosella, corbalan, laural}@lidi.info.unlp.edu.ar

III-LIDI (Instituto de Investigación en Informática LIDI)
Facultad de informática. Universidad Nacional de La Plata.
La Plata. Argentina. 1900

Abstract

In recent years, research on techniques for developing controllers for autonomous robots has been conducted. Evolutionary Algorithms are among the most popular tools used in this type of problem, mostly for its capacity to adapt to the environment. Nevertheless, they are usually applied to produce a controller that will not continue its adjustment after concluding this process. This causes trouble to a controller when it is used in a dynamic environment. In this paper, the combination of a state-of-the-art modular neuro-evolution algorithm with a specific evolutionary algorithm is proposed. The former method is used to generate the controller while the later is used to adjust it during its operation. As a result, an adaptable controller based on a minimal topology neural network is obtained. The method proposed was tested in a goal-reach problem with satisfying results. Finally, conclusions are presented.

Keywords: Modular Evolution, Neural Networks, Evolutionary Algorithms.

Resumen

En los últimos años se han realizado diversas investigaciones en técnicas para el desarrollo de controladores para robots autónomos. Los Algoritmos Evolutivos son una de las herramientas más utilizadas en este tipo de problemas por su capacidad de adaptación al entorno. Sin embargo, en su mayoría, la aplicación se concentra en la fase de generación del controlador no permitiendo realizar adaptaciones posteriormente. Esto perjudica la aplicación del controlador en ambientes dinámicos. Este artículo propone extender la evolución del controlador a lo largo de su vida útil combinando un método basado en evolución de módulos neuronales con un algoritmo evolutivo específico. El primer método es utilizado para producir los controladores mientras que el segundo ajusta al controlador durante su funcionamiento. Como resultado, se obtiene un controlador adaptable en la fase de ejecución basado en una red neuronal de arquitectura mínima. La propuesta de este artículo fue medida en la resolución de problemas del tipo alcance de objetivos con resultados satisfactorios. Finalmente, se exponen las conclusiones.

Palabras Claves: Evolución de Módulos Neuronales, Redes Neuronales, Algoritmos Evolutivos.

[1] Becario III-LIDI. Auxiliar Docente. Facultad de Informática. UNLP.
[2] Becario de Doctorado CONICET. Dirigido por Ing.De Giusti y Lic.Lanzarini. JTP. Fac. Informática. UNLP
[3] Becario Formación Superior UNLP. Dirigido por Ing.De Giusti y Lic.Lanzarini. Auxiliar Docente. Fac. Informática. UNLP
[4] Profesor Titular DE. III-LIDI. Facultad de Informática. UNLP

1. Introduction

Evolutionary Robotics (ER) is a methodology that uses Evolutionary Algorithms (EA) to develop controllers for autonomous robots, usually in the form of Artificial Neural Networks (ANN). Artificial Neural Networks are chosen partly because of their ability to learn and adapt to the environment and partly because they are easily represented in an Evolutionary Algorithm. [11].

Although the development of controllers using this kind of strategies is very useful, it is rarely applied to the adjustment of the controller after it enters in operation. When acting in a dynamic environment, like the real world for example, an autonomous robot must adapt itself to the changing conditions or it will not be able to correctly perform its tasks.

In this paper, the combination of a state-of-the-art modular neuro-evolution algorithm with a specific evolutionary algorithm is proposed. The former method is used to generate controllers while the later is used to adjust them during their operation. As a result, an adaptable controller based on a minimal topology neural network is obtained.

This task is divided into two stages: First, a population of controllers is produced using information from the environment. Then, a small subset of that population is selected and submitted to a second Evolutionary Algorithm inside the robot, allowing it to adjust itself to changes in the environment.

Controllers produced by the first stage consist of a combination of different neural modules that are simpler than the controller they form part of and were evolved previously and independently. This seeks to minimize the time required to obtain an acceptable performance. [5].

If the environment in which the autonomous robot acts does not suffer any changes, the best controller generated by the first stage will be good enough to solve the problem because they have precisely evolved to do it. However, in the real world changes occur all the time: light conditions vary, obstacles are not always in the same place or the target to reach is a different one. Owing to it, the controller must do some adjustments when it is in operation. This is the reason for the second stage: to adapt the autonomous robot interaction with its environment. To do so, a small population formed by the best three controllers found in the first stage is evolved using another Evolutionary Algorithm designed to run inside the robot.

This paper is organized as follows. Section 2 describes the evolutionary algorithm used to produce the initial controllers. Section 3 introduces how the continuous adaptation of the robot to its changing environment is achieved. Section 4 shows the results obtained. Finally, Section 5 presents our conclusions.

2. First Stage Evolutionary Algorithm

In order to solve complex problems, different approaches that divide the original problem into simpler ones have been proposed. Even though the existing methods vary in the way they acquire knowledge, most of them adopt a strategy based on the evolution and combination of different modules.

In this direction, methods that combine Incremental Evolution and NeuroEvolution have been developed to offer adaptive mechanisms that minimize the necessary knowledge needed to obtain a working controller, giving rise to Neural Networks composed by several other networks [1]. As the controller is composed by several modules, it is important to define which module should be active at each time-step [13]. There exist different alternatives: from the use of an ad-hoc designed decision tree [4] to mechanisms that embed the selection into the controller [2].

If each of these simpler tasks is successfully and independently solved, it should be possible to combine these solutions to complete the complex task. On this assumption, an extension to the NeuroEvolution of Augmenting Topologies (NEAT) method which incorporates the concept of modules is used.

It is assumed that there exists a set of Neural Networks in which each of them, called a module, is capable of solving one of the simple tasks. The objective of this stage is to produce a **Unified Neural Network** constituted by the combination of all of these modules and capable of solving the complex task.

A brief description of the original NEAT method alongside the proposed extension will be presented in the following subsections. For more details, refer to [5].

2.1. Standard NEAT

The standard NEAT implementation has been shown to be a highly effective NE method in several domains [8]. It addresses three problems commonly found in ANN systems: 1) how to crossover topologically disparate chromosomes, 2) how to protect new topological innovation, and 3) how to keep topologies as simple as possible throughout evolution [9]. This is accomplished through historical markings, speciation, and incremental complexification.

First, each genome in NEAT includes a list of connection genes, each of which refers to two node genes that are connected. In order to perform a crossover, the system must be able to tell which genes match up between any two individuals in the population. For this reason, NEAT keeps track of the historical origin of every gene. Two genes that have the same historical origin represent the same structure (although possibly with different weights), since they were both derived from the same ancestral gene from some point in the past. Tracking the historical origins requires very little computation. Whenever a new gene appears (through structural mutation), an innovation number is incremented and assigned to that gene. The innovation numbers thus represent a chronology of every gene in the system, and allow the crossover of diverse networks without extensive topological analysis. With historical markings the problem of having to match different topologies [7] is avoided.

Second, NEAT networks are speciated so that individuals compete primarily within their own niche. In this way, topological innovations are given time to optimize their structure before they have to compete with the entire population. Also, networks share the fitness of their species [3], to prevent one species from taking over the entire population.

Third, NEAT networks are built up from a minimal configuration and complexified incrementally to ensure that solutions of minimal complexity are searched first. This procedure has two advantages: First, it minimizes topology bloat, and second, it improves the efficiency of evolution by complexifying the search space only as needed. For more details about NEAT, see Stanley and Miikkulainen [9].

2.2. NEAT with Modules

The incorporation of neural modules to the NEAT method implies carrying out several modifications. The first one is related to the neural networks that compose the initial population. In the original proposal, it is assumed that there is not enough knowledge of the problem to specify the topology of those networks. In addition, starting with minimal networks allows the method to explore simpler solutions first. In this extension, networks solving different parts of the problem are known and it is possible to fill the initial population with variations of a unified neural network. This network is built up from merging each of the available modules within a same structure.

Since the tasks solved by each module are part of a single complex tasks, it is expected that more than one module will use the same inputs or produce the same output. The unified neural network will have the union of the inputs of each module as input. The modules are connected to those inputs without undergoing any modification. The unified network outputs depend on the task to solve, and for this reason the network will have as many output neurons as the problem needs.

More than one module may generate the same output of the network. It is also possible that different modules produce opposite stimuli for similar inputs, since the tasks solved by each of them may be contradictory. To allow the evolution to adjust the contribution of each module to the unified network outputs, rewarding expected responses and making opposite stimuli compatible, each module output neurons become hidden neurons. To each of these converted neurons, a new connection is added that links this neuron to the output neuron that produces the response which was originally yielded by former neuron. The connection is established with a 1.0 weight, so the original stimulus reaches the output neuron without being affected. This new connection is not considered as part of any module, but belongs to a unified neural network. Figure 1 shows the combination process of two modules to produce a unified neural network.

During the building process of the unified network, each connection and neuron integrated into the network is marked with an identifier associated with the module that it belongs to. This is done to simplify the tracking of the modules that compose each network once the evolution has started.

Another proposed modification to NEAT is the way in which genetic operators are applied to produce new genomes. Originally, the mutation operator was in charge of generating innovations, perturbing weights, establishing new connections among existing neurons, or inserting a new neuron after dividing an existing connection.

In this paper, the mutation operator scope has been restricted. It is only possible to modify the weight of a connection, if it did not originally belong to any of the modules making up the network undergoing mutation. In the same way, it is not allowed to establish new connections among neurons of the same original module, being only valid to do so among neurons of different modules. Eventually, it is only possible to add a neuron if an existing connection is previously divided, which, once again, should not be a connection contributed by any of the modules. These restrictions force the evolutionary method to generate the necessary structure to allow the original modules to interact so that they can reach the solution of the posed complex task together.

The rest of the evolving method is not different from standard NEAT; historical markings are kept in the genomes of the population, the original crossover operator is used, and the population is divided into species according to a compatibility criterion, dividing the fitness of each member proportionally to the number of genomes belonging to the same niche.

The reason for which the topology and connection weights of each module cannot be changed is due to the fact that, since these are fixed, the evolutionary algorithm will search in a more reduced space than if it were to do it over an entirely mutable network. This should favor a faster convergence towards better solutions.

It is worth noting that bigger structures could be generated compared with the ones that could be obtained if started from a minimum topology. However, when the difficulty of the task increases, the complexity of the neural network proportionally increases, and generating a structure that acts as an interface between the modules is simpler than solving the whole problem.

3. Second Stage Evolutionary Algorithm

The second stage of the evolution takes place on the autonomous robot and it should run for an indefinite period as long as the controller is operational. On this stage, a small population consisting of three controllers generated in the first stage is maintained. The selection of the controllers that belong to the initial population may be accomplished in several ways: The three best controllers of the entire population of the first stage can be chosen to integrate the new population. Alternately, the fittest controller of each of the three most performing species may be selected. In this work, the first strategy was adopted. The selected controllers provide the genetic pool that will allow the second stage algorithm to adapt the robot behavior when changes in its environment occur.

The Evolutionary Algorithm running on this stage works by performing small modifications in the controllers, causing variations in its behavior. To do so, each controller in the current population is evaluated and a new controller is produced using the two best controllers found. This new controller replaces the worst one in the current population. This guaranties that the most performing controller will never be lost.

An Extended Linear Recombination operator applied to the best controllers found is used to produce the new controller. This operator was selected based on the results obtained in [10]. Let P1 and P2 be the selected controllers to be used in the recombination and O the new controller obtained by applying the Extended Linear Recombination operator to them. Equation 1 shows how O is generated.

$$O_i = P_{1i} + \frac{s_i \cdot r_i \cdot a \cdot \left(P_{2i} - P_{1i}\right)}{\left\|P_1 - P_2\right\|} \quad (1)$$

Sub-index $i \in [1, N]$ represents each of the network's weights and N, the number of weights. a defines the step size to use, and its value is computed as 2^{-ku}. k determines the precision of such step, taking values between 4 and 20 as suggested in [6]. $a \in [0,1]$ is a uniformly distributed random number, $r_i \in [-0.5,0.5]$ is a uniformly distributed random number and it represents the maximum variation that can appear between parent and offspring in the *i-th* weight. s_i is -1 or $+1$, random uniform. *s* is the direction of the step.

Inside this possible area the offspring are not uniform at random distributed. The probability of creating an offspring near the parents is high. Only with low probability offspring are created far away from the parents. For more details on this operator, see [6].

4. Results and Discussion

This work proposal can be used to produce a controller based on Evolutionary Artificial Neural Networks that is capable of guiding a Khepera II robot to reach a certain target while avoiding obstacles. The controller must also be able to adapt to changes in its environment when it enters in operation.

As only the robot proximity and light sensors were used, the target that the robot must reach is actually a light source located at random places inside a maze. The goal of the robot is to navigate freely without crashing until it approaches a bright zone where it will try to stay.

The initial population of controllers was generated according to the methodology explained in section 2: Two independently produced modules were combined: one of them gives the robot the capacity to avoid obstacles while the other seeks for the nearest light source. Each of these modules contains a Recurrent Artificial Neural Network generated using conventional NEAT as described in

section 2.1. Figure 1 shows the network obtained from merging the two modules. Both modules use the same input data from the proximity sensors. The outputs of each module were combined using two newly added neurons that commands the motors. This Unified Neural Network has 16 input neurons, 8 from each one of the proximity sensors and 8 from each one of the light sensors, and has 2 output neurons, each one connected to each of the motors.

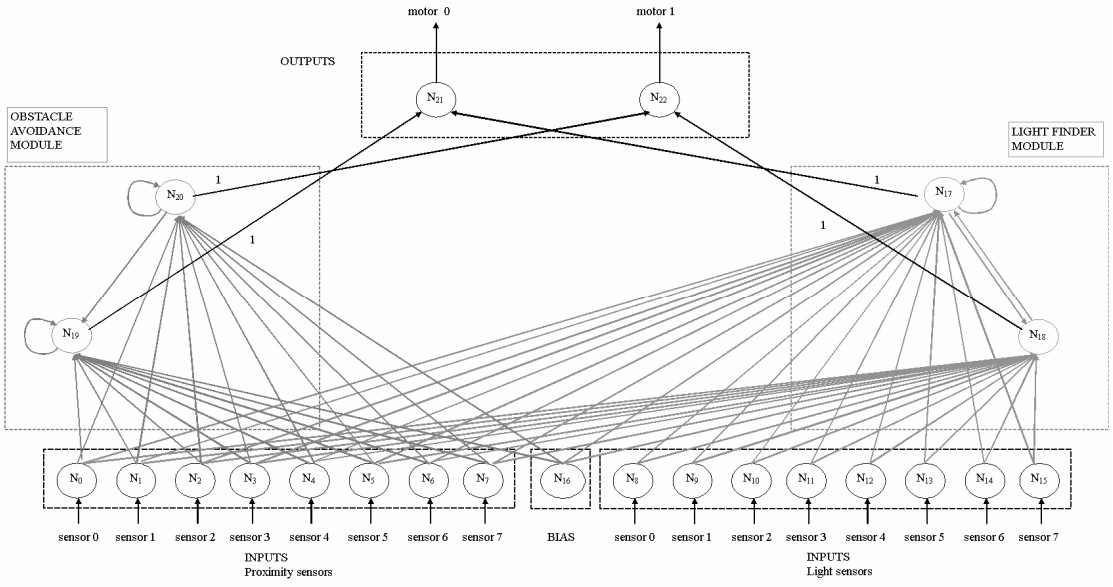


Figure 1

Slightly mutated copies of the original Unified Neural Network genome were used to form the initial population for the first stage. This population was evolved using *NEAT with modules* in the manner explained in section 2.2. In this stage, the fitness of a controller was computed according to its capacity to combine the original modules, as expressed by equation 2.

$$fitness = coef_{obs} \times Fit_{Obs} + coef_{lig} \times Fit_{Lig} \quad (2)$$

In equation 2, Fit_{Obs} and Fit_{Lig} are the fitness of each controller measured according to the fitness functions of the obstacle evasion problem and the light reaching problem, respectively. $coef_{obs}$ and $coef_{lig}$ are constants used to regulate the importance of each behavior in the resultant fitness value. In this work, 1 and 1.3 were respectively used to put more emphasis on the light reaching part.

Figure 2 depicts the fittest Unified Neural Network obtained as a result of the evolutionary process.

The best three performing controllers produced by the first stage were selected to constitute the initial population of the second stage algorithm. These controllers were continuously evolved as long as the robot was operative. A reduced population size was chosen to minimize the computational time required to evaluate the fitness of the population. Each of the three controllers was independently evaluated from the others when controlling the real robot.

After the evaluation of the population was completed, the controllers were sorted according to their fitness. The best two controllers were selected to produce a new one using the Extended Linear Recombination operator. This operator takes two parents and produces an offspring similar to the best parent, but changed in direction of the other parent. The newly produced controller replaced the worst controller of the population.

The frequency in which this process has to be done depends on how much the environment changes. If there are no changes, it may even not be necessary to do it at all. Nevertheless, it is advisable to run the algorithm in the first generations to allow the robot to adapt to the differences in the simulator compared to the real world.

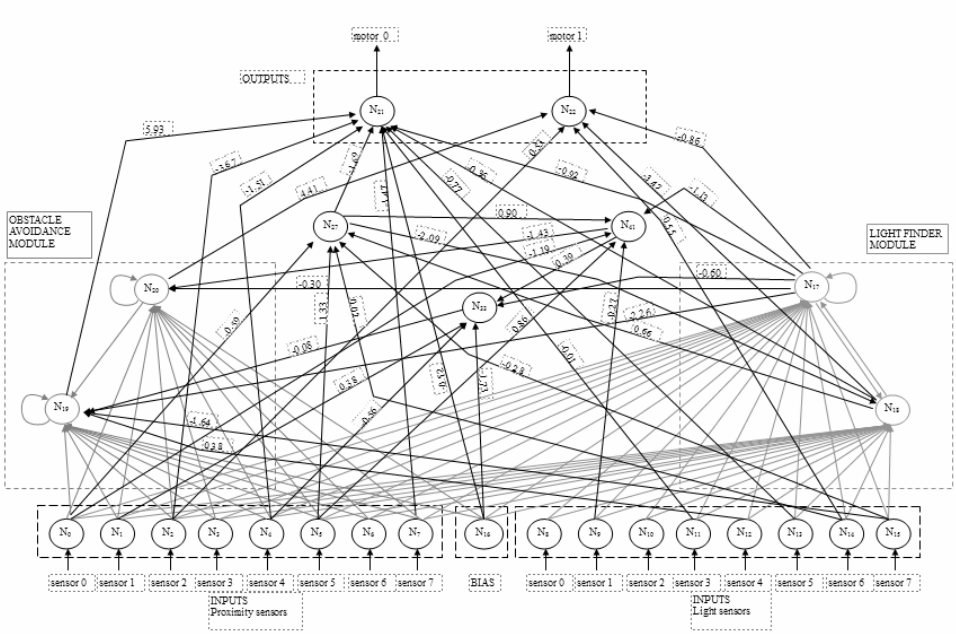


Figure 2

Tests to analyze the ability of the robot to adapt to changes in the environmental conditions were conducted. Figure 3 shows the mean fitness of two different robots generation by generation: one using the second stage algorithm (solid line) and the other not using it (dashed line). In the 60th generation, a sudden change in the environmental conditions was introduced and maintained until the 225th generation.

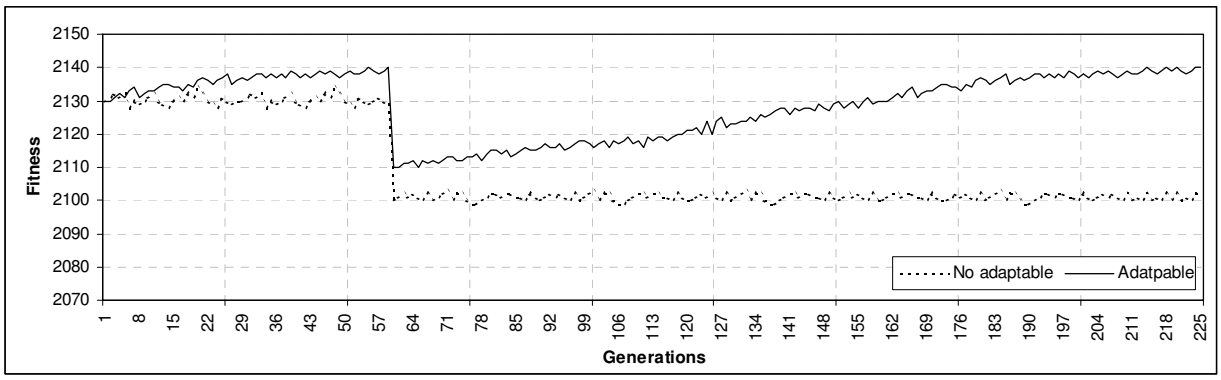


Figure 3: Mean fitness of the adaptable (solid line) and non-adaptable (dashed line) robots.

In the first 60 generations, an improvement in the fitness of the adaptable controller can be noticed compared to the non-adaptable controller. Fitness variations on both lines are due to different conditions in which the controllers were evaluated. This explains why the non-adaptable controller varies its fitness even if the controller does not change.

After the environment changed in generation 60, a drop in the fitness value of both controllers can be seen. This is because none of the controllers were appropriated for the new environmental conditions. However, as the second stage evolutionary algorithm of the adaptable robot continued to adjust its controllers, an improvement in its fitness value can be noticed. After generation 200, its fitness is almost as good as it was in the previous environment. On the other hand, the non-adaptable robot did not improve at all.

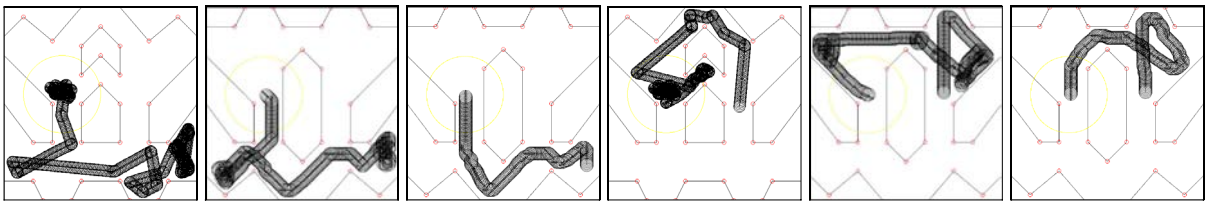


Figure 4: Performance of two adaptable robots in different periods: a) and d) in the original environment, b) and e) immediately after the changes, and c) and f) when the adjustment occurred.

5. Conclusions

Evolutionary Algorithms, though capable of providing excellent results in several areas, have one unfavorable characteristic: the adaptation process may be slow and costly in time for some problems. Because of that, it is rarely employed after an acceptable solution is found.

This paper applies a strategy that reduces the required time to find a suitable controller, taking advantage of the combination of simple neural modules, based on recurrent neural networks generated independently of the problem to solve. Furthermore, a second strategy allows the autonomous robot to continue its adaptation to changes in the environment after entering in operation.

The continuous evolution of a controller on a Khepera II robot capable of reaching a certain target while avoiding obstacles has proved to be successfully in adapting to changes in its environment. Adaptation is a crucial task for an autonomous robot and it may be the difference between a successful controller and a failed one.

References

[1] Bruce, J. Miikkulainnen, R. Evolving Populations Of Expert Neural Networks. Department of Computer Sciences, The University of Texas at Austin. Proceedings of the Genetic and Evolutionary Computation Conference. (GECCO-2001, San Francisco, CA), pp. 251-257. 2001.

[2] Corbalán L., Osella Massa G., Lanzarini L., De Giusti A. ANELAR. Arreglos Neuronales Evolutivos de Longitud Adaptable Reducida. X Congreso Argentino de Ciencias de la Computación. CACIC 2004. Universidad Nacional de La Matanza. Bs.As. Argentina. ISBN 987-9495-58-6. October 2004.

[3] Goldberg, D.E., Richardson, J. Genetic algorithms with sharing for multimodal function optimization. pp. 148-154. 1987.

[4] Olivera J. Lanzarini, L. Cyclic Evolution. A new strategy for improving controllers obtained by layered evolution. VI Workshop de Agentes y Sistemas Inteligentes 2005. Concordia, Entre Ríos. Argentina. ISBN: 950-698-166-3. October 2005.

[5] Osella Massa G., Vinuesa H., Lanzarini L. Modular Creation of Neuronal Networks for Autonomous Robot Control. 3rd IEEE Latin American Robotics Symposium. LARS 2006. Chile. October 2006.

[6] Pohlheim H. Evolutionary Algorithms: Overview, Methods and Operators. GEATbx version 3.7. Genetic and Evolutionary Algorithm Toolbox for use with Matlab. pp: 28-29. November 2005.

[7] Radcliffe, N.J.: Genetic set recombination and its application to neural network topology optimization. Neural computing and applications 1. pp. 67-90. 1993.

[8] Stanley, K.O., Miikkulainen, R. Competitive coevolution through evolutionary complexification. Journal of Artificial Intelligence Research 21. 2001.

[9] Stanley, K.O., Miikkulainen, R. Evolving neural networks through augmenting topologies. Evolutionary Computation 10. pp. 99-127. 2002.

[10] Vinuesa, H. Lanzarini, L. Neural Networks Elitist Evolution. 29th Internacional Conference Information Technology Interfaces (ITI 2007). Dubrovnik. Croatia. 2007.

[11] Walker J. Garrett, S. Wilson, M. The Balance Between Initial Training and Lifelong Adaptation in Evolving Robot Controllers. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 36, NO. 2. APRIL 2006.

[12] Yao, X. Liu, Y. Ensemble Structure of Evolutionary Artificial Neural networks. Computational intelligence Group, School of Computer Sciencie University College. Australian Defence Force Academy, Canberra, ACT, Australia 2600. 1996.

[13] Yao, X. Evolving Artificial Neural networks. School of Computer Sciencie The University of Birmingham Edgbaston, Birmingham B15 2TT. Proceedings of the IEEE. Vol.87, No.9, pp.1423-1447. September 1999.

Towards a Comparison Criteria for $\mathcal{C}DeLP$

María Laura Cobo* **Guillermo Ricardo Simari**
Artificial Intelligence Research and Development Laboratory (LIDIA)
Universidad Nacional del Sur
Bahía Blanca, B8000CPB, Argentina
[mlc,grs]@cs.uns.edu.ar

Abstract

The development of systems with the ability to reason about change notion and actions has been of great importance for the artificial intelligence community. The definition and implementation of systems capable of managing defeasible, incomplete, unreliable, or uncertain information has been also an area of much interest. With a few exceptions research on these two ways of reasoning was independently pursued. Nevertheless, they are complementary and closely related, since many applications that deal with defeasible information also depends on the occurrence of events and time.

$DeLP$ is an argumentative system appropriate for commonsense reasoning. The defeasible argumentation basis of $DeLP$ allows to build applications that deal with incomplete and contradictory information in dynamic domains. Thus, the resulting approach is suitable for representing agent's knowledge and for providing an argumentation based reasoning mechanism for that agent (see for example [6, 1]). It is interesting to extend this system adding mechanisms to manage events and time as $\mathcal{C}DeLP$ [7]. Here we analyze how to develop a comparison criteria for arguments built up from causal information and considers *commonsense rules of inertia*.

Key words: Argumentative Systems, Knowledge Representation, Defeasible Reasoning, Commonsense Reasoning, Temporal Reasoning, Reasoning about change notion and actions.

1 Introduction

The development of systems with the ability to reason about causal information (information inferred from events occurrence), has been of great importance for the artificial intelligence community. Research in this area has been interested on this issue as a way to find a solution

*Partially supported by Agencia de Promoción Científica y Tecnológica (PICT 13096, PICT 15043, PAV 2003 Nro. 076) and the Universidad Nacional del Sur

for a wider variety of problems where the occurrence of actions and the moment they occur makes a difference [9].

Argumentation Systems [5, 17], were developed in order to deal with incomplete or unreliable information. In real scenarios this situation is quite common, specially when we deal with dynamic systems, *i.e.*, systems where the knowledge available to reason with changes frequently (new information become available or information that used to be available became unavailable or invalid). Usually, since it is very difficult to represent all the information related to the objects under consideration, the information that appears as supporting our reasoning is incomplete. As a matter of fact, there are formalisms such as the *Situation Calculus* [20] where this problem is quite relevant. When new information about an entity becomes available, *i.e.*, knowledge changes, we must revise all the representation.

In order to improve on this problem we will follow an *argumentative* approach. $DeLP$ [8] is an argumentation system, which combines results from Logic Programming and Defeasible reasoning providing tools for knowledge representation and commonsense reasoning. We are interested in the development of an argumentative system based on $DeLP$ that can deal with causal information, a first approach leads us to basics definition of $\mathcal{C}DeLP$ [7]. This extension introduces the concept of causal information, *i.e.* information *events* and/or *time* dependant.

This work takes $\mathcal{C}DeLP$ basic definition and analyzes some difficulties in arguments comparison criteria, since $DeLP$ criteria is not proper enough to solve some new situations, particularly observable from inertia chains and actions occurrences.

In some literature actions and events are consider as different things, it is necessary then to clarify that in the context of the present work this terms are considered as synonyms.

The paper is structured as follows, on Section 2 a refresh of basic $\mathcal{C}DeLP$s definitions is made. Then a criteria for arguments comparison, taking in consideration only the strong information of each argument, is presented. Later on, on Section 4 we establish a way to compare looking only at inertial chains over arguments. On Section 5 we show some examples where difficulties to compare arguments from the amount and kind of actions, each argument uses, are emphasized. The last section presents the conclusions.

2 $\mathcal{C}DeLP$ Definition

In a previous work [7] we presented a new argumentation language based on a clausal language *Event Calculus* [10, 22, 14] called $\mathcal{C}DeLP$. The basics of this language depend on the main predicates of a simple version of *Event Calculus*, (*SEC*), which are:

$happens(E,T)$:	E takes place on T.
$holdsAt(F,T)$:	F holds at T.
$initiates(E,F,T)$:	F starts to hold after E, and is not freed on $T+1$.
$terminates(E,F,T)$:	F ceases to hold after E at T.
$releases(E,F,T)$:	F is not subject to inertia after E at T
$initiallyP(F)$:	F holds form time zero.

where E represents events, T time moments and F fluents. The calculus complete axiomatization depends on the chosen time ontology. For example if we consider a discrete ontology,

we can use the ontology presented by Mueller [15], or the more complete ontology from Miller and Shanahan research [14].

Once we have the syntax of the literals of the language, we can define it:

DEFINITION **1** ($\mathcal{C}DeLP$)

A $\mathcal{C}DeLP$ program is defined in terms of three disjoint sets: a set Π of *facts* and *strict rules*, a set Δ of *defeasible rules*, and a set Υ of *inertial defeasible rules*, where

- A *fact* is a literal, *i.e.*, a ground atom, o a negated ground atom.
- A *strict rule* is a rule denoted as "$Head \leftarrow Body$", where $Head$ is a literal and $Body$, is finite set of literals. A strict rule can also be written as: $L_0 \leftarrow L_1, \ldots, L_n (n > 0)$, where each $L_i, i \geq 0$ is a literal.
- A *defeasible rule* is a rule noted as $L_0 \prec L_1, \ldots, L_n$. Again L_i is a literal and $i \geq 0$
- A *inertial defeasible rule* is a *defeasible rule* that denotes some fluent inertia.

Consequently, this modification causes a change on the definition of argument:

DEFINITION **2** *($\mathcal{C}DeLP$ Causal Argument Structure)*

Let L be a literal and $\mathcal{P} = (\Pi, \Delta, \Upsilon)$ a $\mathcal{C}DeLP$ program. We say that $\langle \mathcal{A}, \mathcal{B}, L \rangle$ is a causal argument structure for L, if $\mathcal{A}$ is a set of defeasible rules from Δ and $\mathcal{B}$ is a set of defeasible rules form Υ, such that $\mathcal{A} \cup \mathcal{B}$ verifies $DeLP$ argument structure definition [8].

$$\Delta_1 = \left\{ \begin{array}{l} holdsAt(flies(X), T) \prec holdsAt(bird(X), T), \\ holdsAt(flies(X), T) \prec \sim holdsAt(injured(X), T), holdsAt(bird(X), T) \\ \sim holdsAt(flies(X), T) \prec holdsAt(injured(X), T) \\ \sim holdsAt(injured(X), T+5) \prec holdsAt(injured(X), T), \end{array} \right\}$$

$$\Upsilon_1 = \left\{ \begin{array}{l} holdsAt(injured(X), T+1) \prec holdsAt(injured(X), T), \\ \sim holdsAt(injured(X), T+1) \prec \sim holdsAt(injured(X), T), \end{array} \right\}$$

$\langle \mathcal{D}, \mathcal{I} \sim holdsAt(flies(tina), 5) \rangle$ is the proper definition of argument $\mathfrak{A}_1$ for fluent $\sim holdsAt(flies(tina), 5)$, where:

$$\mathcal{D} = \left\{ \sim holdsAt(flies(tina), 5) \prec holdsAt(injured(tina), 5), \right\}$$

$$\mathcal{I} = \left\{ \begin{array}{l} holdsAt(injured(tina), 5) \prec holdsAt(injured(tina), 4), \\ holdsAt(injured(tina), 4) \prec holdsAt(injured(tina), 3), \\ holdsAt(injured(tina), 3) \prec holdsAt(injured(tina), 2), \\ holdsAt(injured(tina), 2) \prec holdsAt(injured(tina), 1), \\ holdsAt(injured(tina), 1) \prec holdsAt(injured(tina), 0) \end{array} \right\}$$

In previous work [7] the comparison criteria among argument was pendant. In the following sections we will discuss several aspects the criteria should take on consideration. At the same time some previous definition to achieve this goal will be presented.

3 Comparing arguments from the base

In order to decide when an argument is better than other, we must design a comparison criteria that allow the proper choice of the wining argument. As a first step, this criteria should take on consideration the argument base, *i.e.* the subset of Π that origins the defeasible reasoning that takes place after that. This subset may be formed by fluents ($holdsAt(f,t)$ kind of formula) or sentences that defines events occurrence ($happens(a,t)$ kind of formula). This last kind of formula clearly represents actions that actually take place or are granted to happen through strong information use only.

Several aspects should be taken in consideration about the type of information that appear at the tree's base, for example we must evaluate if the same fluent, on different moments, appears in the base an root of the tree; or the amount of events the argument considers, as well as possible repetitions of the same event. At this point is necessary to point out that all the events considered are facts or can be proven only form information in the Π set of the program, this means that are always 'strong' information. This status of actions makes an argument with more actions more plausible.

The fact that the same fluent is in the base and the root may imply the use of an inertial rule in a direct way or that its state is a precondition to further gain of other fluents. As a first approximation lets consider the simple example that follows:

EXAMPLE **1** Let P_1 be a $CDeLP$ program defined as: $P_1 = (\Pi_1, \Delta_1, \Upsilon_1)$, where:

$$\Pi_1 = \left\{ \begin{array}{l} holdsAt(ligh\_off, 0) \\ happens(switch, 1) \\ happens(switch, 3) \\ happens(switch, 5) \\ holdsAt(ligh\_on, T) \leftarrow \sim holdsAt(light\_off, T) \\ holdsAt(ligh\_off, T) \leftarrow \sim holdsAt(light\_on, T) \end{array} \right\}$$

$$\Delta_1 = \left\{ \begin{array}{l} holdsAt(light\_on, T+1) \prec holdsAt(light\_off, T), happens(switch, T+1) \\ holdsAt(light\_off, T+1) \prec holdsAt(light\_on, T), happens(switch, T+1) \end{array} \right\}$$

$$\Upsilon_1 = \left\{ \begin{array}{l} holdsAt(light\_off, T+1) \multimap holdsAt(light\_off(X), T), \\ holdsAt(light\_on, T+1) \multimap holdsAt(light\_on(X), T), \end{array} \right\}$$

Form P_1 we can built the arguments shown in figure 1. In this example, we clearly see that

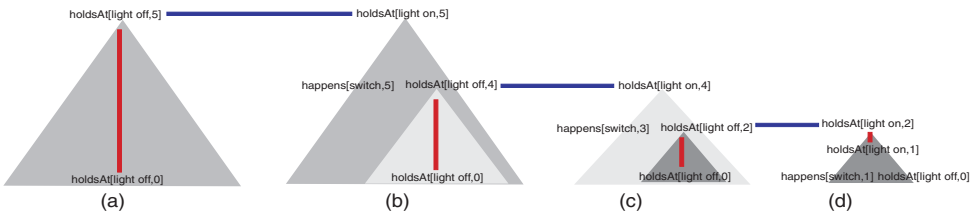


Figure 1: reasoning line

the first argument only appeals to the use of an inertial rule to warrant $holdsAt(light\_on, 5)$

(this fact makes this argument very weak in a discussion since it will be very easy to defeat); the other argument also uses an inertial rule, but the chain is shorter and it uses event's occurrence, so clearly is a better supporting argument than the first one.

If we take a look to the arguments against $holdsAt(light\_on, 5)$ we can see we count with two as well. In this case both arguments uses inertial rules and the happening of actions. We can also see that the event is the same, but since the happening of events is part of environment check (part of Π) the argument with more events is clearly more specific and more adjusted to reality.

Looking at the previous example we can draw definitions, in order to formalize the ideas shown there. One of them decide when two arguments are incomparable (from base), informally this situation takes place if we can not decide which one is better for the basic information they use to build up their conclusion. Formally this notion is captured on definition 3

DEFINITION **3** *(incomparable from base)*

Suppose we have arguments $\mathfrak{A}_1$ and $\mathfrak{A}_2$. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ the set of fluents and $\mathcal{E}_1$ and $\mathcal{E}_2$ the set of events (both $\in \Pi$) such that are the basics of each argument. We state that the arguments are **incomparable from base** if:

$$\text{If } \mathcal{F}_1 \cap \mathcal{F}_2 = \emptyset \text{ and,} \qquad \text{If } \mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$$

As we explain before the amount of events that take place are important in order to determine the strength of an argument, particularly when the event that actually happens is the same over and over again, as in the example 1 where the action that happens is $switch$ obviously on different moments. We can consider events repetition as a sign of strength and define a criteria that considers this:

DEFINITION **4** *(repeated events)*

Suppose we are comparing two arguments $\mathfrak{A}_1$ and $\mathfrak{A}_2$, such that are not *incomparable*. And some action a such that $happens(a, T) \in \mathcal{E}_1 \cap \mathcal{E}_2$ if $happens(a, T_2) \in \mathcal{E}_1$ but $\notin \mathcal{E}_1 \cap \mathcal{E}_2$ with $T_2 \neq T$ Then $\mathfrak{A}_1$ is stronger than $\mathfrak{A}_2$

or we can extend the criteria not analyzing the kind of event but only the amount like in the following definition:

DEFINITION **5** *(preferring from the base)*

Suppose we are comparing two arguments $\mathfrak{A}_1$ and $\mathfrak{A}_2$, such that are not *incomparable*.

$\mathfrak{A}_1$ is going to be preferred over $\mathfrak{A}_2$ if the following condition holds:

1. $\mathfrak{A}_1$ uses more observed events than $\mathfrak{A}_2$. Formally: $|\mathcal{E}_1 - (\mathcal{E}_1 \cap \mathcal{E}_2)| = n$, $|\mathcal{E}_1 - (\mathcal{E}_1 \cap \mathcal{E}_2)| = m$ with $n > m$.

2. $\mathfrak{A}_1$ uses more observed fluents than $\mathfrak{A}_2$. Formally: $|\mathcal{F}_1 - (\mathcal{F}_1 \cap \mathcal{F}_2)| = n$, $|\mathcal{F}_1 - (\mathcal{F}_1 \cap \mathcal{E}_2)| = m$ with $n > m$.

Observed fluent or event are those that can be inferred only form information available in the Π set of the corresponding program.

4 Comparing arguments that uses inertia

If conflict appears through the use of inertial chains that are comparable, then the shorter one should be preferred since it has less points of attacks, making the argument stronger in certain sense.

In order to make an analysis over inertial chains, we must take in consideration the subset of Υ used in both arguments (the inertial rules used on the construction of the arguments). If the intersection of those sets is non empty then, both uses the same inertial rule. Being the case, we should prefer the argument with the shorter inertial chain for that fluent.

DEFINITION **6** *(inertial chains length)*

For some argument structure $\mathfrak{A} = \langle \delta, \upsilon, L \rangle$ and some literal l granted through an inertial chain. Then an inertial rule, $i \in \upsilon$, is used several times. i has the form $holdsAt(l,t) \multimap body$, where $body$ is a proper body for the rule. The length of the chain is evaluated like this:

Let T_{max} be the maximal time point of the inertial chain and let T_{min} be the minimal value of parameter t in the argument structure such that:

$$\forall m \; T_{min} < m < T_{max}[holdsAt(l,t) \multimap body \in \upsilon]$$

and there is no other time points $x > T_{max}, y < T_{min}$ that verifies the previous condition. Then the *length of the inertial chain* is $T_{max} - T_{min}$

DEFINITION **7** *(analyzing inertial chains)*

Suppose we are comparing two arguments $\mathfrak{A}_1$ and $\mathfrak{A}_2$, such that are not *incomparable*. Let $\mathfrak{A}_1 = \langle \delta_1, \upsilon_1, L \rangle$ and $\mathfrak{A}_2 = \langle \delta_2, \upsilon_2, L \rangle$. If $\upsilon_1 \cap \upsilon_2 \neq \emptyset$ (both arguments uses inertia over the same fluent) then, we should consider the following algorithm:

```
no argument is marked for each rule in the intersection do
while only one or none is marked do
  take the literal on the head of the rule
  get the length of inertial rule from both arguments
  compare lengths
  mark the argument with shorter inertial chain length
If both are marked then return blocking situation
  else return as preferred argument the marked one
```

Notice that can be the case that we can decide that neither are preferable since more than one inertial chains are applied and the shorter one is not in the same argument.

If we consider example 1 we can see that among the fact that the argument that ends to be preferred form the base (has more events), that same argument is preferred under definition above. Looking at figure 1 argument (a) has an inertial chain of length 5 for fluent $light\_off$, while the one for argument (b) is 4, 2 for argument (c) and finally 1 for argument (d).

Let us illustrate a blocking situation through a very well known example in frame problem literature, the *Yale Shooting Problem*. This problem presents a scenario where the necessity of the inertial rules on the specification of causal information is clear. In this sense we

are going to use Steve Hanks and Drew McDermott [9] problem to show a pair of arguments that are defeated by a mutual blocking status.

EXAMPLE **2** Let consider the following program $\mathcal{P}_2 = (\Pi_2, \Delta_2, \Upsilon_2)$, where:

$$\Pi_2 = \left\{ \begin{array}{l} initiallyP(alive(fred)) \\ initiallyP(moving(fred)) \\ happens(load(gun), 0) \\ happens(shoot(gun), 2) \\ \\ holdsAt(loaded(gun), T) \leftarrow happens(load(gun), T) \\ holdsAt(moving(X), T) \leftarrow happens(startMovement(X), T) \\ \sim holdsAt(moving(X), T) \leftarrow happens(endMovement(X), T) \\ \\ holdsAt(alive(X), T) \leftarrow holdsAt(alive(X), T_1), \sim happens(shoot(Y), T_1), T = T_1 + 1 \end{array} \right\}$$

$$\Delta_2 = \left\{ \begin{array}{rcl} \neg holdsAt(alive(X), T) & \prec & holdsAt(loaded(Y), T), happens(shoot(Y), T) \\ holdsAt(alive(X), T) & \prec & holdsAt(moving(X), T), happens(shoot(Y), T) \end{array} \right\}$$

$$\Upsilon_2 = \left\{ \begin{array}{l} holdsAt(loaded(Y), T+1) \multimap holdsAt(loaded(Y), T) \\ \sim holdsAt(loaded(Y), T+1) \multimap \sim holdsAt(loaded(Y), T) \\ holdsAt(moving(X), T+1) \multimap holdsAt(moving(X), T) \\ \sim holdsAt(moving(X), T+1) \multimap \sim holdsAt(moving(X), T) \end{array} \right\}$$

If we take a look to arguments in figure 2 we see that no choice among them is possible.

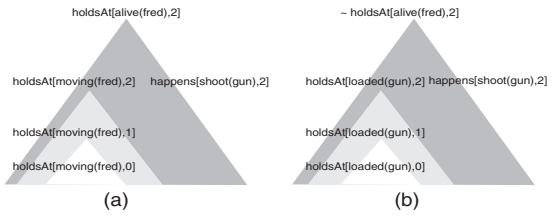


Figure 2: complementary arguments that leads to blocking defeat

Since both arguments are structurally identical, both uses a chain of inertial rules but over different fluents, independent fluents *i.e.* not comparable from the information available in the program.
If we eliminate $intiallyP(moving(fred))$ from set Π, and then we add $happens(startMovement(fred), 1)$ the arguments shown in figure 2 will look like this: In this case, the argument (a) has de shorter inertial chain than the argument shown in argument (b). If we decide to choose this measure as a way to avoid blocking then argument (a) should be preferred; but we can avoid the fact that this is the only fact that favors one of the arguments because looking at the base both sets are not comparable.

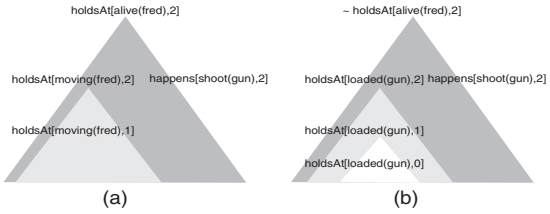


Figure 3: complementary arguments

5 Analyzing 'defeasible' events

There may events that are not registered in Π in a direct way, *i.e.* their real happening may depend on fluents that are granted in a defeasible way (*'defeasible' events*). If this is the case, we must analyze the supporting information for that event to take place, for granting the hole argument. If this is the case, the analysis of the base is not enough an we must take a look to the inner nodes of the tree in order to detect this kind of situation.

In general fluents are properties that holds on certain time point, *i.e.* properties that are either true or false at any time (*'strong' fluents*). With the introduction of defeasibility we lead to a situation where some fluents are *granted* or not. This means that we can build a defeasible proof for the fluent, since the defeasible nature of the proof a fluent granted in this way has less strength (from a knowledge perspective) than 'strong' fluents. So the following definition is needed:

DEFINITION **8** *Defeasible fluents*

Let $\mathcal{F}$ be a fluent. $\mathcal{F}$ is a defeasible fluent if its status is granted through an argument structure that can not be defeated.-

Let see an example where event occurrence depend on defeasible fluents:

EXAMPLE **3** Imagine an scenario where certain alarm clock must ring at some determine moment, only if it has enough battery. Here is the program, $\mathcal{P}_3 = (\Pi_3, \Delta_3, \Upsilon_3)$, where:

$$\Pi_3 = \left\{ \begin{array}{l} initiallyP(asleep(fred)) \\ initiallyP(enough\_battery) \\ happens(clock\_rings, T) \leftarrow holdsAt(enough\_battery, T) \\ \\ \sim holdsAt(asleep(fred), T) \leftarrow happens(clock\_rings, T) \end{array} \right\}$$

$$\Delta_3 = \left\{ \begin{array}{l} \sim holdsAt(enough\_battery, T) \prec \quad holdsAt(enough\_battery, T_1), T = T_1 + 10) \\ \end{array} \right\}$$

$$\Upsilon_3 = \left\{ \begin{array}{l} holdsAt(asleep(X), T+1) \multimap holdsAt(asleep(X), T) \\ \sim holdsAt(asleep(X), T+1) \multimap \sim holdsAt(asleep(X), T) \\ holdsAt(enough\_battery, T+1) \multimap holdsAt(enough\_battery, T) \end{array} \right\}$$

Taking a look to arguments, we can see that $clock\_rings$ depends on fluent $enough\_battery$ to take place. This fluent can or can not be granted in a defeasible way. Can be granted

through an inertial rule an can not by a defeasible rule that captures the notion of information's ageing (more details can be found at [7]). Clearly in this case the argument that uses the defeasible rule beats the one that uses de inertial rule. Situation like the one in figure 3

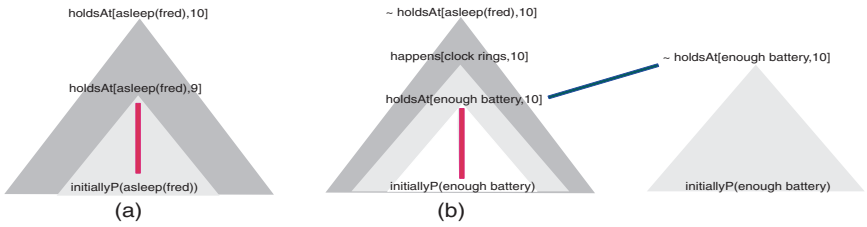


Figure 4: events that depend on defeasible fluents

are solved through the following definition:

DEFINITION **9** *(Inertia, Actions and Ageing)* Suppose we are comparing two arguments $\mathfrak{A}_1$ and $\mathfrak{A}_2$, and some fluent f such that $holdsAt(f,t)$ is granted on both arguments if:

1. there is a rule $i \in \mathfrak{A}_1$ such that forms an inertial chain that granted f at other time point T; and there is a rule $r \in \mathfrak{A}_2$ such that is an *information ageing rule* or an *event rule* such that grants $\sim f$ at the same T, or

2. there is a rule $a \in \mathfrak{A}_1$ such that is an *information ageing rule* such that grants f; and there is a rule $e \in \mathfrak{A}_2$ such that is an *event rule* such that grants f at the same T

then $\mathfrak{A}_2$ is preferred over $\mathfrak{A}_1$

It must be taken in consideration that an *information ageing rule* is a sentence $a \in \Delta$ of the form $head \prec body$ where $body$ does not include literals like $happens(action, time)$ and must have $head$ at a previous moment. On the other hand an *event rule* is also a sentence $\in \Delta$ also of the form $head \prec body$ but in this case $body$ must include a literals like $happens(action, time)$. As example of information ageing rule we can take $\sim holdsAt(enough\_battery, T) \prec holdsAt(enough\_battery, T_1), T = T_1 + 10)$ while $holdsAt(enough\_battery, T) \prec happens(replace\_battery, T)$ is an event rule.

Clearly we can consider the possibility of changing the battery, modifying the program on example 3. Leading to the same program with the addition of the boxed sentence.

$$\Pi_3 = \left\{ \begin{array}{l} initiallyP(asleep(fred)) \\ initiallyP(enough\_battery) \\ happens(clock\_rings, T) \leftarrow holdsAt(enough\_battery, T) \\ \boxed{happens(replace\_battery, T) \leftarrow \sim holdsAt(enough\_battery, T)} \\ \sim holdsAt(asleep(fred), T) \leftarrow happens(clock\_rings, T) \end{array} \right\}$$

$$\Delta_3 = \left\{ \begin{array}{l} \neg holdsAt(enough\_battery, T) \prec holdsAt(enough\_battery, T_1), T = T_1 + 10) \\ \boxed{holdsAt(enough\_battery, T) \prec happens(replace\_battery, T)} \end{array} \right\}$$

$$\Upsilon_3 = \left\{ \begin{array}{l} holdsAt(asleep(X), T+1) \multimap holdsAt(asleep(X), T) \\ \sim holdsAt(asleep(X), T+1) \multimap \sim holdsAt(asleep(X), T) \\ holdsAt(enough\_battery, T+1) \multimap holdsAt(enough\_battery, T) \end{array} \right\}$$

In this case we can see that the wining argument is build due to the occurrence of new event introduced. Figure 3 illustrate this new argument.

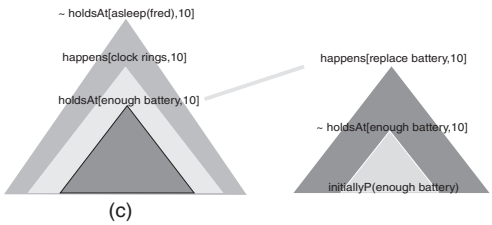


Figure 5: additional argument

Other situation that follows from the previous one, is what happens if the action under consideration can only takes place once, like $born$. This is so because a certain individual can $born$ only once. In other to deal with this kind of events we need to introduce axioms on set Π of the program that sets the uniqueness on action occurrence and use *Temporal disagreement* notion as well (see definition 10).

DEFINITION **10** *(Temporal Disagreement)*

Two arguments $\mathcal{A}_1^{t_1} = \langle \mathcal{A}_1, (l_1, t_1)) \rangle$, $\mathcal{A}_2^{t_2} = \langle \mathcal{A}_1, (l_2, t_2)) \rangle$ are in *temporal disagreement* of there are time points t_1, t_2 such that $\mathcal{A}_1^{t_1}$, $\mathcal{A}_2^{t_2}$ are in disagreement. Meaning that: $\Pi \cup \{(l_1, t_1), (l_2, t_2)\} \vdash \bot$

Let see an example where this kind of disagreement actually happens.

EXAMPLE **4** Imagine an scenario where certain alarm clock must ring at some determine moment, only if it has enough battery. Here is the program, $\mathcal{P}_4 = (\Pi_4, \Delta_4, \emptyset)$, where:

$$\Pi_4 = \left\{ \begin{array}{rl} \sim happens(born(baby), T) \leftarrow & happens(born(baby), T_1), T > T_1 \\ \sim happens(fall\_umbilical\_cord(baby), T) \leftarrow & happens(fall\_umbilical\_cord(baby), T_1), \\ & T > T_1 \\ & \\ happens(born(mary's\_baby), T) \leftarrow & holdsAt(at\_term, T) \\ happens(born(mary's\_baby), T) \leftarrow & holdsAt(before\_term, T) \\ & \\ happens(fall\_umbilical\_cord(mary's\_baby), T) \leftarrow & happens(born(baby, T_1), \\ & holdsAt(before\_term, T_1), T = T_1 + 10 \\ happens(fall\_umbilical\_cord(mary's\_baby), T) \leftarrow & happens(born(baby, T_1), \\ & holdsAt(at\_term_1), T = T_1 + 7 \\ & \\ happens(become\_pregnant(mary), 1) & holdsAt(first\_baby(mary), T) \\ holdsAt(calm\_person(mary), T) & \end{array} \right\}$$

$$\Delta_4 = \left\{ \begin{array}{rl} holdsAt(at\_term, T) \prec & happens(become\_pregnant(mary), T_1), \\ & holdsAt(calm\_person(mary), T_1), T = T_1 + 40) \\ holdsAt(before\_term, T) \prec & happens(become\_pregnant(mary), T_1), \\ & holdsAt(first\_baby(mary), T_1), T = T_1 + 38) \end{array} \right\}$$

We can observe that these arguments are very similar, their base has the same number of sentences and they are of the same kind also. At first sight it seems to be no conflict among them, but both appearing events can only happens once. If we choose to keep with the first happening of the event (over the time line) then for event $fall\_umbilical\_cord(baby)$ we must choose argument (b), but for event $born(baby)$ the sub-argument of (a) must win.

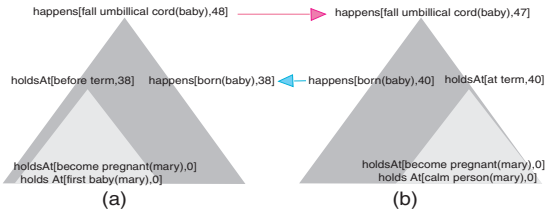


Figure 6: unique events on conflict

Although these considerations are extremely important for the preference criteria, the addition of these facts to a proper definition is in current analysis.

6 Conclusions and Future work

Argumentative systems such as $DeLP$ have been significant for the evolution of commonsense reasoning area. We have already presented $\mathcal{C}DeLP$ as an alternative argumentative system that captures today's definition of this kind of reasoning (*i.e* reason with events, time, cause-effect principle).

$\mathcal{C}DeLP$, differentiates two kind of defeasible rules, the ones already available at plain $DeLP$ and the ones that captures inertia over fluents truth value. These modifications make urgent a change in preference criteria among arguments. In that sense we present several definitions to solve some of the critical points.

A complete definition of the comparison criteria is needed. The analysis must take in consideration how the use of inertial rules are used in the construction of arguments and counterarguments. Clearly, an argument that uses inertial rules is weaker than an argument that uses only non-inertial defeasible rules. Definitions and examples presented in this work offers a solution to these items. Nevertheless, some other questions remain. We must define how to determine preference when 'defeasible' events are used on arguments. In which order we should apply the definitions presented in this work and the remaining ones in order to get better results?

References

[1] CAPOBIANCO, M., CHESÑEVAR, C. I., AND SIMARI, G. R. argumentation and the dynamics of warranted beliefs in changing environments. *Autonomouns Agents and Multi-Agent Systems 11*, 2 (2005), 127–151.

[2] CARTHY, J. M. First order theories of individual concepts and propositions. In *Machine Intelligence 9*, B. Meltzer and e. D. Michie, Eds. Edinburgh Univerity Press, Edinburgh, 1979, pp. 120–147.

[3] CARTHY, J. M. Applications of Circunscription to Formalizing commonsense Knowledge. *Artificial Intelligence 28* (1986), 89–116.

[4] CARTHY, J. M., AND HAYES, P. J. Some philosophical problems from the standpoint of artificial intelliegence. In *Machine Intelligence 4*, B. Meltzer and e. D. Michie, Eds. Edinburgh Univerity Press, Edinburgh, 1969, pp. 463–502.

[5] CHESÑEVAR, C. I., MAGUITMAN, A. G., AND LOUI, R. Logical models of argument. *ACM Computing Surveys 4*, 32 (2000), 337–383.

[6] CHESÑEVAR, C. I., SIMARI, G. R., GODO, L., AND ALSINET, T. Expansion operators for modelling agent reasoning in possiblistic defeasible programming. In *EUMAS* (2005), pp. 474–475.

[7] COBO, M. L., AND SIMARI, G. R. Combining a causal language with argumentation: A first approach. In *Proceedings of the XII Congreso Argentino de Cs. de la Computación, CACIC06* (San Luis, 2006), pp. 1231–1242.

[8] GARCIA, A. J., AND SIMARI, G. R. Defeasible logic programming: an argumentative approach. *TPL 4* (2004), 95–138.

[9] HANKS, S., AND DERMOTT, D. M. Nonmonotonic logic and temporal projection. *Artificial Intelligence 33* (1987), 379–412.

[10] KOWALSKI, R., AND SERGOT, M. A logic-based calulus of events. *New Generation Computing 4*, 1 (1986), 67–895.

[11] LLOYD, J. W. *Foundations of Logic Programming*, third edition ed. Springer-Verlag, New York, 1995.

[12] MCCARTHY, J. A Form of Non-Monotonic Reasoning. *Artificial Intelligence 13* (1980), 27–39.

[13] MCDERMOTT, D., AND DOYLE, J. Nonmonotonic logic 1. *Artificial Intelligence 13* (1980), 41–72.

[14] MILLER, R., AND SHANNAHAN, M. Some alternative formulations of the event calculus. *Computational Logic: Logic Programming and Beyond 14* (2004), 703–730.

[15] MUELLER, E. T. Event calculus reasoning through satisfiability. *Journal of Logic and Computation* (2002), 452–490.

[16] MUELLER, E. T. *Commonsense Reasoning*. Morgan Kaufman an imprint of ELsevier, 2006.

[17] PRAKKEN, H., AND VREESWIJK, G. Logics for defeasible argumentation. In *Handbook of Philosophical Logic*, D. G. (ed.), Ed. Kluwer Academic Publishers, 1998.

[18] PRIOR, A. *Past, Present and Future*. Clarendon Press, 1967.

[19] REITER, R. A Logic for Default-Reasoning. *Artificial Intelligence 13* (1980), 81–132.

[20] REITER, R. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.

[21] RESCHER, N., AND URQUHART, A. *Temporal Logic*. Springer-Verlag, 1971.

[22] SHANAHAN, M. Representing continuous change in the event calculus. In *Proceedings ECAI 90* (1990), pp. 598–603.

[23] VAN EMDEM, M., AND KOWALSKI, R. The Semantics of Predicate Logic as a Programming Language. *Journal of the Association for Computing Machinery 23*, 4 (1976), 733–742.

Aprendizaje de Juegos mediante Cúmulos de Partículas con Tamaño de Población Variable

Victoria Leza
Facultad de Informática. Universidad Nacional de La Plata
La Plata, Argentina, 1900
victorialeza@gmail.com

Laura Lanzarini
III-LIDI (Instituto de Investigación en Informática LIDI)
Facultad de Informática. Universidad Nacional de La Plata
La Plata, Argentina, 1900
laural@lidi.info.unlp.edu.ar

Abstract

Game learning has encouraged the development of several Artificial Intelligence strategies. Even though there exist different approaches, the adaptation to the information environment is an expected characteristic when we are to solve a complex problem, since it is unnecessary to count with the codification of any type of initial knowledge.

In this area, neural networks have provided excellent results though with a generally high computational cost. There exists previous work in which population heuristics have been used to reduce the training times.

In particular, the use of evolving strategies based on fixed-size particle swarms has rendered successful results. This paper proposes to modify the size of the population during the adaptation process, adding and deleting individuals in function of their skills to solve the problem.

The proposed method has been applied in a neural networks training to play the game TicTaeToe. The results obtained have been compared to those of a strategy based on particle swarms with fixed-size population. Finally, conclusions and some future lines of work are presented

Keywords: Swarm Intelligence, Particle Swarm Optimization, Neural Networks,

Resumen

El aprendizaje de juegos ha motivado el desarrollo de numerosas estrategias de Inteligencia Artificial. Si bien existen diferentes enfoques, la adaptación al entorno de información es una característica deseable cuando se busca resolver un problema complejo ya que hace innecesaria la codificación de cualquier tipo de conocimiento inicial.

En esta área, las redes neuronales han brindado excelentes resultados pero generalmente con un costo computacional elevado. Existen trabajos previos donde se han utilizado heurísticas poblacionales, para reducir el tiempo de entrenamiento.

En particular, el uso de estrategias evolutivas basadas en cúmulos de partículas de tamaño fijo ha producido resultados satisfactorios. Este artículo propone modificar el tamaño de la población durante el proceso de adaptación, agregando y eliminando individuos en función de su aptitud para resolver el problema planteado.

El método propuesto ha sido aplicado en el entrenamiento de una red neuronal para jugar el juego de TaTeTi. Los resultados obtenidos han sido comparados con los de una estrategia basada en cúmulos de partículas con tamaño de población fija. Finalmente se exponen las conclusiones así como algunas líneas de trabajo futuras.

Palabras Claves: Optimización mediante Cúmulos de Partículas (PSO), Redes Neuronales

1. Introducción

Desde los principios de la Inteligencia Artificial, los investigadores han intentado desarrollar estrategias que les permitan a las computadoras aprender de una manera automática. En general se busca crear algoritmos capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos. De esta manera, a semejanza del aprendizaje de los seres humanos, para aprender no se requiere conocer la solución específica del problema sino que la misma es extraída de la información disponible. Si bien estas estrategias poseen una amplia gama de aplicaciones, el aprendizaje de distintos tipos de juegos ha sido motivo de inspiración continua y aun lo sigue siendo.

Muchos algoritmos de juegos tradicionales basan su funcionamiento en un árbol de jugadas. Esta estructura les permite seleccionar el próximo movimiento a realizar. Sin embargo, más allá de las complicaciones que implica la construcción del árbol y los esfuerzos realizados tanto para reducir su tamaño como para mejorar las estrategias sobre él aplicadas, resultan inapropiados cuando el espacio de búsqueda es amplio y complejo.

Como alternativa a este tipo de soluciones se propone el uso de estrategias inteligentes que permitan desarrollar agentes capaces de aprender un juego basados únicamente en sus reglas y teniendo conocimiento del estado actual del mismo.

En esta dirección existen soluciones basadas en Redes Neuronales que han producido excelentes resultados pero generalmente con un costo computacional elevado. Existen trabajos previos donde se han utilizado heurísticas poblacionales, para reducir el tiempo de entrenamiento. En particular, la optimización mediante cúmulos de partículas (PSO - Particle Swarm Optimization) ha producido resultados satisfactorios en una amplia gama de problemas de optimización, incluyendo el entrenamiento de redes neuronales y la minimización de funciones. En todos los casos, el tamaño de la población utilizada permanece fijo durante el proceso adaptativo.

Este artículo se propone diseñar un agente basado en una red neuronal entrenada mediante cúmulos de partículas con tamaño de población variable donde los individuos serán incorporados y eliminados en función de su aptitud para resolver el problema planteado. Esto se realiza a través del concepto de edad que permite determinar el tiempo de vida de cada elemento de la población. Si bien existen distintas formas para estimar el tiempo que cada individuo permanecerá dentro de la población, se utilizará el definido en [Lan01] ya que ha demostrado ser capaz de considerar adecuadamente la aptitud de los individuos.

Con la intención de no poblar excesivamente un mismo lugar del espacio de soluciones, se utilizará una técnica de nichos de construcción dinámica definida en [Bir06]. En ella, se realiza un análisis estadístico de la población para definir la distancia adecuada entre dos individuos que pertenecen al mismo nicho.

Este artículo se encuentra organizado de la siguiente forma: …

2. Algoritmos basados en Cúmulos de Partículas

Un algoritmo basado en Cúmulos de Partículas, también llamado Particle Swarm Optimization (PSO), es una técnica heurística poblacional donde cada individuo representa una posible solución del problema y realiza su adaptación teniendo en cuenta tres factores: su conocimiento sobre el entorno (su valor de fitness), su conocimiento histórico o experiencias anteriores (su memoria), el conocimiento histórico o experiencias anteriores de los individuos situados en su vecindario [Ken95]. Su objetivo es evolucionar su comportamiento de manera de asemejarse a los individuos con más éxito dentro de su entorno. En este tipo de técnica, cada individuo permanece en continuo

movimiento dentro del espacio de búsqueda y nunca muere. Por su parte, la población puede verse como un sistema multiagente donde cada individuo o partícula se mueven dentro del espacio de búsqueda guardando y eventualmente comunicando, la mejor solución que han encontrado.

Existen distintas versiones de PSO; las más conocidas son *gBest PSO* que utiliza como criterio de vecindad a la población completa y *lBest PSO* que, por el contrario, utiliza un tamaño de vecindad pequeño [Ken95][Shi99]. El tamaño de la vecindad influye en la velocidad de convergencia del algoritmo así como es la diversidad de los individuos de la población. A mayor tamaño de vecindad, la convergencia del algoritmo es más rápida pero la diversidad de individuos es menor.

Cada partícula p_i está compuesta por tres vectores y dos valores de fitness:

- El vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ almacena la posición actual de la partícula en el espacio de búsqueda.
- El vector $pBest_i = (p_{i1}, p_{i2}, \ldots, p_{in})$ almacena la posición de la mejor solución encontrada por la partícula hasta el momento.
- El vector velocidad $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ almacena el gradiente (dirección) según el cual se moverá la partícula.
- El valor de fitness $fitness\_x_i$ almacena el valor de aptitud de la solución actual (vector x_i).
- El valor de fitness $fitness\_pBest_i$ almacena el valor de aptitud de la mejor solución local encontrada hasta el momento (vector $pBest_i$)

La posición de una partícula se actualiza se la siguiente forma

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (1)$$

Como se explicó anteriormente, el vector velocidad se modifica teniendo en cuenta su experiencia y la de su entorno. La expresión es la siguiente:

$$v_{ij}(t+1) = w \,.\, v_i(t) + \varphi_1 \,.\, rand_1 \,.\, (pBest_i - x_i(t)) + \varphi_2 \,.\, rand_2 \,.\, (g_i - x_i(t)) \quad (2)$$

donde w representa el factor de inercia [Shi98], φ_1 y φ_2 son constantes de aceleración, *rand1* y *rand2* son valores aleatorios pertenecientes al intervalo (0,1) y g_i representa la posición de la partícula con el mejor *pBest_fitness* del entorno de p_i (*lBest* o *localbest*) o de todo el cúmulo (*gBest* o *globalbest*). Los valores de w, φ_1 y φ_2 son importantes para asegurar la convergencia del algoritmo. Para más detalles sobre la elección de estos valores puede consultar [Cle02] y [Ber02].

La figura 1 contiene el algoritmo PSO básico

```
S ← InicializarCumulo()
while no se alcance la condición de terminación do
    for i = 1 to size(S)  do
       evaluar la partícula xi del cúmulo S
       if fitness(xi) es mejor que fitness(pBesti) then
             pBesti ← xi; fitness(pBesti) ← fitness(xi)
        end if
    end for
    for i = 1 to size(S)  do
        Elegir gi según el criterio de vecindad utilizado
        vi ← w . vi + φ1 . rand1 . (pBesti – xi) + φ2 . rand2 . (gi – xi)
        xi ← xi + vi
    end for
end while
Salida : la major solución encontrada
```

Figura 1

3. Cúmulos de Partículas con tamaño de población variable

El algoritmo descripto en la sección anterior trabaja sobre una población de tamaño fijo. En este artículo se propone utilizar un cúmulo de tamaño variable buscando mejorar la relación de compromiso existente entre la velocidad de convergencia y la diversidad de la población.

La variación en el tamaño de la población se logra permitiendo que las partículas se reproduzcan y mueran durante el proceso de adaptación. Esto requiere la definición de dos criterios: uno que permita saber si un individuo existente puede generar nuevos descendientes y otro que indique como quitar elementos de la población.

3.1. Criterio de reproducción

La reproducción busca incrementar la velocidad de convergencia incorporando individuos en las zonas menos pobladas. Debe ser aplicada con moderación a fin de no aumentar excesivamente la cantidad de individuos dentro del espacio de búsqueda. Por lo tanto, es preciso analizar la distribución de las partículas dentro de dicho espacio para luego determinar si la reproducción puede llevarse a cabo o no.

Para calcular la probabilidad de reproducción de cada una de las n partículas de la población S, se analizará su entorno tomando un radio de distancia r calculado de la siguiente forma [Bir06]:

$$d_i = \min\{\| x_i - x_j \| \; ; \forall j \; x_i, x_j \in S \; \wedge \; x_i \neq x_j\} \qquad \text{con } i = 1..n \tag{3}$$

$$r = \frac{\sum_{i=1}^{n} d_i}{n} \tag{4}$$

El valor de r a utilizar es el mismo para todas las partículas.

La probabilidad de reproducción de la i-ésima partícula se calcula de la siguiente forma [Fer05]:

$$Prob_i = 1 - (CantVecinosExistentes_i \; / \; MAX\_CANTIDAD) \quad \text{con } i = 1..n \tag{5}$$

siendo *CantVecinosExistentes* la cantidad de partículas encontradas dentro del entorno del i-ésimo individuo de la población S y *MAX_CANTIDAD* es un parámetro del algoritmo que representa la máxima cantidad de individuos permitida.

Para aquellos casos en que la medida de distancia de (3) no resulte apropiada para la representación del espacio de búsqueda seleccionada, pueden verse otras alternativas en [Ert02].

Cada individuo de la población, según su probabilidad de reproducción, podrá generar un único descendiente aplicando mutación uniforme sobre los pesos de la red neuronal y copiando en el nuevo individuo su vector velocidad y su conocimiento social.

3.2. Cálculo del tiempo de vida

La permanencia de los individuos dentro de la población depende de su tiempo de vida. Dicho valor se expresa en cantidad de iteraciones, transcurridas las cuales, la partícula es eliminada. Este valor tiene una estrecha relación con la aptitud de la partícula y permite que los mejores individuos permanezcan en la población por mayor tiempo, influenciando el comportamiento del resto.

Para estimar el tiempo de vida de cada individuo de la población de utilizó el método de asignación por clases definido en [Lan01] ya que ha demostrado poseer la capacidad de brindar buenos resultados con una cantidad de individuos muy inferior a la empleada por los métodos convencionales.

En [Lan01] los individuos de la población son agrupados según su valor de aptitud en *k* clases utilizando un método de clustering competitivo del tipo *winner-take-all*, por ejemplo k-medias.

Sobre el resultado de este agrupamiento puede aplicarse uno de los siguientes métodos:

a) Asignación de tiempo de vida fijo por clase

Se divide el máximo tiempo de tiempo de vida a asignar por la cantidad de clases, *k*. Esto permite saber el rango de tiempo que le corresponde a cada clase. Dentro de una misma clase, sus individuos recibirán un tiempo de vida proporcional a la clase a la que pertenecen y a la cantidad de individuos que se encuentran en su misma clase, de la siguiente forma:

```
AnchoClase := MAX_LT / k
TVClasesAnt := (ClaseMasCercana – 1)* AnchoClase
TVClaseActual  := AnchoClase
Desplazamiento  = ( fitness[i] – Clase[ClaseMasCercana].MinFit ) /
             abs(Clase[ClaseMasCercana].MaxFit – Clase[ClaseMasCercana].MinFit))
TiempoDeVida[i] := trunc(TVClasesAnt + WidthClass * Desplazamiento)
```

donde

- *AnchoClase* es el rango del tiempo de vida asignado a cada clasen (dada por MAX_LT / Número de clases)
- *ClaseMasCercana* es el número de clase a la que pertenece cada individuo.
- *TVClaseActual* es el rango de tiempo de vida de la clase a la pertenece el individuo.
- *TVClasesAnt* es el rango de tiempo de vida asignado a las clases anteriores a *ClaseMasCercana.*
- Clase[*ClaseMasCercana*].MinFit y Clase[*ClaseMasCercana*].MaxFit son los valores de aptitud mínimo y máximo de la clase a la que pertenece el individuo en consideración.
- *Fitness[i]* es el valor de aptitud del i-ésimo individuo de la población.

b) Asignación de tiempo de vida proporcional a la cantidad de individuos de cada clase

Cada clase recibe un rango de tiempo de vida proporcional a la cantidad de elementos que contiene. Es decir, que los individuos pertenecientes a las clases numerosas podrán tener un rango de tiempo de vida más amplio. El cálculo es el siguiente:

```
TotalAnt := 0
for i:=1 to ClaseMasCercana-1 do TotalAnt := TotalAnt + Clase[i].Cantidad
TVAnterior := MAX_LT * TotalAnt / TotalIndiv
TVClaseActual  := MAX_LT * Clase[ClaseMasCercana].Cantidad/TotalIndiv
Desplazamiento  = ( fitness[i] – Clase[ClaseMasCercana].MinFit ) /
             abs(Clase[ClaseMasCercana].MaxFit – Clase[ClaseMasCercana].MinFit))
TiempoDeVida[i] := trunc(TVAnterior + WidthClass * Desplazamiento)
```

donde

- *Clases[ClaseMasCercana].Cantidad* representa la cantidad toal de individuos de la clase más cercana.
- *TotalIndiv* es la cantidad total de individuos de la población.

Estas dos formas de calcular los tiempos de vida de los individuos deben combinarse a fin de lograr una asignación correcta. Se propone aplicar la asignación b) durante un cierto porcentaje de la cantidad de generaciones máximas del algoritmo y en las restantes utilizar a). Esto se debe a que los agrupamientos iniciales se realizan sobre individuos que aun no están lo suficientemente adaptados y por consiguiente dan lugar a agrupamientos de tamaños muy disímiles. Si sobre estos

agrupamientos se aplicara directamente la distribución indicada en a) muchos individuos recibirían tiempos de vida similares llevando al algoritmo a incrementar innecesariamente la cantidad de individuos de la población.

3.3. Algoritmo propuesto

El algoritmo comienza con una población de N individuos generados al azar dentro del espacio de búsqueda y calcula para cada uno de ellos su fitness y tiempo de vida correspondientes.

Una vez medida la aptitud de todos los individuos de la población se calcula el radio según la ecuación (4). Esta medida será utilizada para estimar la cantidad de vecinos de cada individuo y tendrá una fuerte incidencia en la probabilidad de reproducción de cada uno de ellos ya que se utilizará la ecuación (5).

La inercia utilizada para actualizar los vectores velocidad es ajustada durante el algoritmo de la siguiente forma [Mei02]:

$$w = w_{start} - \frac{(w_{start} - w_{end})}{ITERACIONES\_TOTALES}.IteracionActual \quad (6)$$

donde w_{start} es el valor inicial de w y w_{end} es el valor final.

El uso de un peso de inercia variable facilita la adaptación de la población. Un valor de w alto al comienzo de la evolución le permite a las partículas realizar movimientos grandes ubicándose en distintas posiciones del espacio de búsqueda. A medida que avanza el número de iteraciones, el valor de w se reduce permitiéndoles realizar un ajuste más fino.

El cálculo del valor de fitness depende del tipo de problema a resolver y será explicado con detalle en la sección de resultados.

En cada iteración y utilizando la probabilidad de reproducción de cada individuo, existe la posibilidad de que se incorporen nuevos. Si esto ocurre, se define una nueva red neuronal copiando la red del padre (individuo original) con una mínima mutación uniforme y manteniendo el mismo vector velocidad.

Finalmente, el algoritmo termina cuando se cumple una de las siguientes condiciones:

- Se alcanzó la cantidad máximas de iteraciones indicadas inicialmente
- El mejor individuo no se ha modificado durante el 20% de las iteraciones totales
- La población se ha quedado sin individuos.

La figura 2 contiene un pseudocódigo del algoritmo descripto.

4. Resultados

El algoritmo propuesto en este trabajo fue utilizado para entrenar una población de redes neuronales para jugar el juego de TaTeTi.

Es decir que cada individuo es una red feedforward totalmente interconectada con 9 entradas, una para cada posición del tablero e igual número de salidas. La función de transferencia utilizada en la capa oculta y en la capa de salida fue la sigmoidal, con la distinción que las salidas fueron redondeadas a modo de que pudiera interpretarse como una variable booleana. Se utilizó una codificación especial para los valores de entrada para permitirle a la red interpretar cuáles son sus posiciones ocupadas, cuáles las libres y cuáles las ocupadas por su oponente en el tablero, estando cada uno representado con el valor 1, 0 y -1, respectivamente. Sus pesos fueron adaptados mediante

el proceso evolutivo. Las mediciones realizadas acerca de la cantidad de neuronas de la capa oculta demuestran que los mejores resultados se obtienen utilizando entre 5 y 7 unidades.

```
// --- Población Inicial ---
S ← {}
for i = 1 to N  do
   x_i ← Red Neuronal inicializada al azar con pesos ∈ (-1,1).
   v_i ← inicializar al azar entre el rango mínimo y máximo especificado.
   evaluar el fitness de la partícula x_i
   pBest_i ← x_i
   fitness(pBest_i) ← fitness(x_i)
   S ← S ∪ { (x_i, v_i, pBest_i, fitness(x_i), fitness(pBest_i)) }
end
Calcular el Tiempo de Vida de cada partícula según 3.2.b)

w ← INERCIA_MAXIMA
while no se alcance la condición de terminación do
    Calcular el radio r según la ecuación (4)
    for i = 1 to size(S)  do
       Elegir LBest_i,la partícula con mejor fitness del entorno de x_i
       v_i ← w . v_i + φ1 . rand_1 . (pBest_i - x_i) + φ2 . rand_2 . (LBest_i - x_i)
       x_i ← x_i + v_i
       Prob_i ← Calcular la probabilidad de reproducción de x_i
       //-- Reproducción --
       if rand < Prob_i then
           Obtener x'_i aplicando mutación uniforme sobre x_i
           Calcular el fitness de x'_i
           S ← S ∪ { (x'_i, v_i, pBest_i, fitness(x'_i), fitness(pBest_i)) }
            if (IteracionActual >  al 20% de las ITERACIONES_TOTALES))
              Calcular el Tiempo de Vida de x'_i según 3.2.a)
            Else
              Calcular el Tiempo de Vida de x'_i según 3.2.b)
        end if

        Disminuir en 1 el tiempo de vida de x_i
        Si Tiempo_de_vida(x_i)=0 then
               S ← S - { (x_i, v_i, pBest_i, fitness(x_i), fitness(pBest_i)) }
    end for
    w ← modificar dinámicamente la inercia
end while
```

Figura 2

Para evaluar el fitness de cada partícula se la hace jugar 7 veces contra otros individuos de la población elegidos al azar. En cada una de estas competencias también se selecciona al azar la partícula que comienza el juego. Cada red recibirá como resultado de cada enfrentamiento un puntaje que se irá acumulando a lo largo de las 7 pruebas. Los valores asignados son: 1 si gana el partido, -2 si pierde y 0 si empata. Aquellas situaciones en las que el juego no se haya resuelto en un número máximo de jugadas indicado a priori, serán consideradas equivalentes a haber perdido, es decir, que el puntaje asignado será -2. Esto último se relaciona con la aplicación de este algoritmo a otro tipo de juego de tablero distintos del TaTeTi. En este caso particular el límite de jugadas impuesto para este caso es el doble del tamaño del tablero, es decir, 18.

Los errores cometidos por la red durante el entrenamiento son considerados de manera aislada. Se considera error a aquellas situaciones en las que la red no brinda una jugada válida ya sea porque sugiere jugar en una posición ocupada o bien porque no indica ninguna salida. En estos casos se suman al total de errores de la res los valores 0.025 y 0.5 respectivamente.

En resumen, la función de fitness utilizada es la siguiente

$$1 / (1 + (\text{CantDeJuegosJugados} + \text{ValorDelContadorDePuntaje}) + \text{ErroresDeLaRed}) \qquad (7)$$

donde *CantDeJuegosJugados* vale 7, *ValorDelContadorDePuntaje* es el resultado acumulado de los 7 juegos efectuados y *ErroresDeLaRed* es el valor acumulado de los errores producidos durante los 7 enfrentamientos.

Se realizaron distintas pruebas utilizando una cantidad máxima de 300 iteraciones porque en ninguno de los casos se observaron mejoras significativas en el fitness más allá de ese valor.

Los valores de aprendizaje cognitivo y social utilizados, φ_1 y φ_2 descriptos en la ecuación (2), fueron ambos en 0.05. Los valores de inercia entre 0.05 y 0. El uso de valores tan pequeños lentifica el movimiento de las partículas impidiendo que se dispersen y caigan en la reproducción excesiva.

La cantidad de vecinas máximas permitidas para la reproducción es 2. Las pruebas demuestran que, con hasta 3 vecinos el crecimiento se mantiene controlado. Sin embargo, un excesivo números de individuos no garantiza alcanzar el óptimo sino más bien, lentificar el algoritmo por lo que el número de vecinos permitidos fue 2.

La cantidad de clases utilizadas para calcular el tiempo de vida de los individuos fue 4. Se realizaron pruebas con valores entre 2 y 10 confirmando que un valor alto para el número de clases, mejora la distinción del fitness entre los individuos pero incrementa sensiblemente el tamaño de la población. Por el otro lado, si la cantidad de clases es muy baja, la población puede llegar a desaparecer. Un valor de 4 clases es adecuado para el problema que se busca resolver.

En la figura 3 pueden observarse los resultados alcanzados con un rango de velocidades mínimo de -0.05 y máximo de 0.05 y una probabilidad de mutación igual a 0, para evitar el movimiento excesivo y, por consiguiente, la superpoblación como se ha mencionado. Como puede verse, las soluciones basada en cúmulos de partículas (PSO) con tamaño de población variable permiten obtener redes con un fitness más alto que *gBest PSO* y *lBest PSO*. También se observa que PSO variable logra valores de fitness equivalentes a sus pares de población fija en un número de generaciones mucho menor.

Las figuras 4 y 5 muestran la evolución de la diversidad de la población para las soluciones basada en cúmulos de partículas (PSO) con tamaño de población variable mostrando que están lejos de la convergencia prematura.

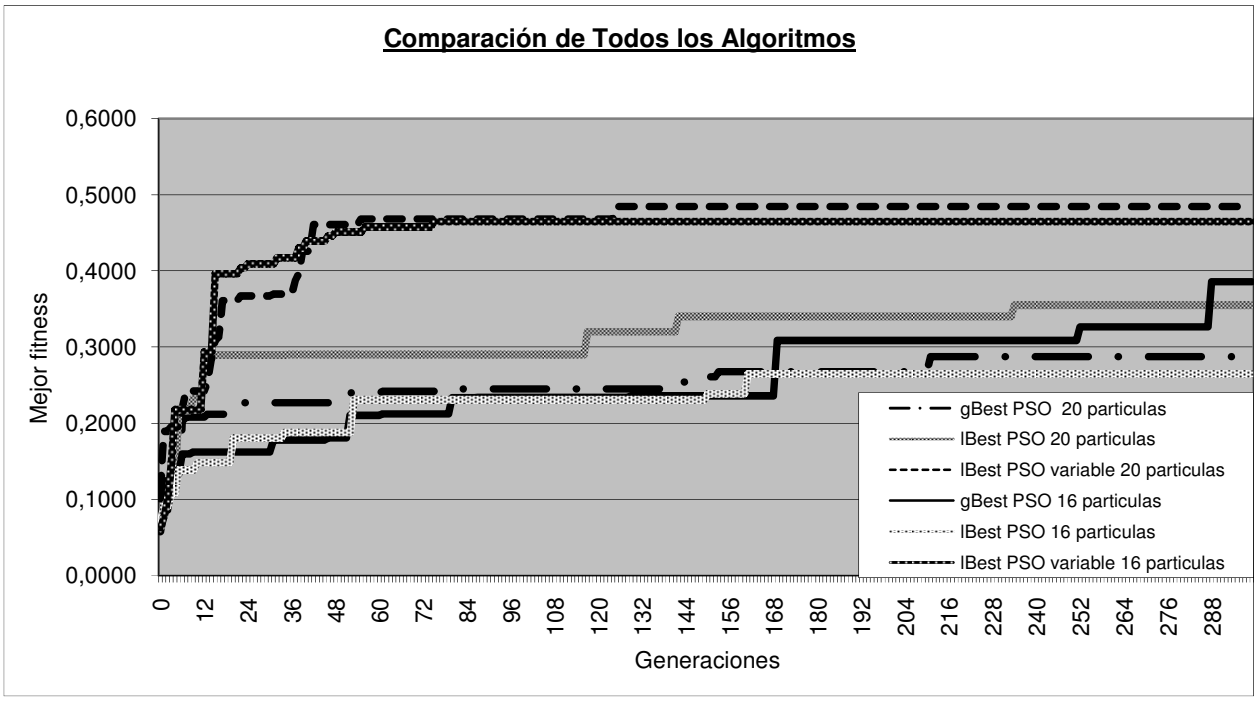


Figura 3

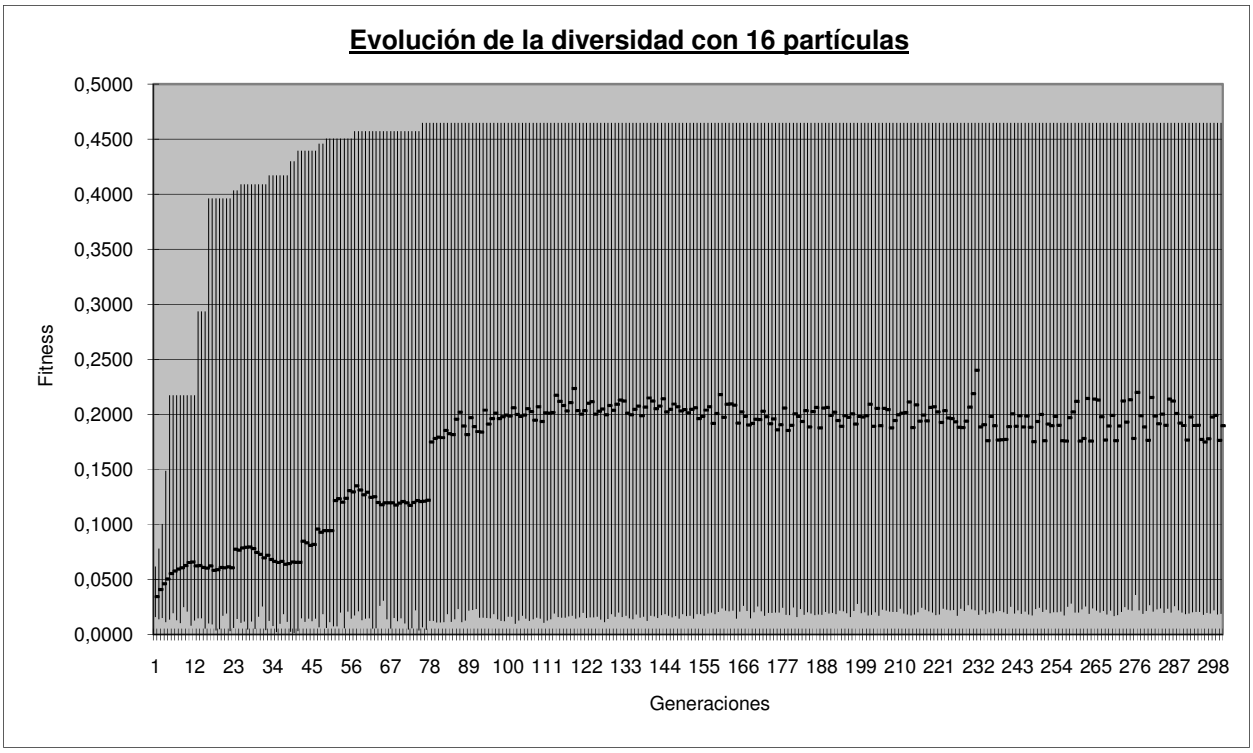


Figura 4

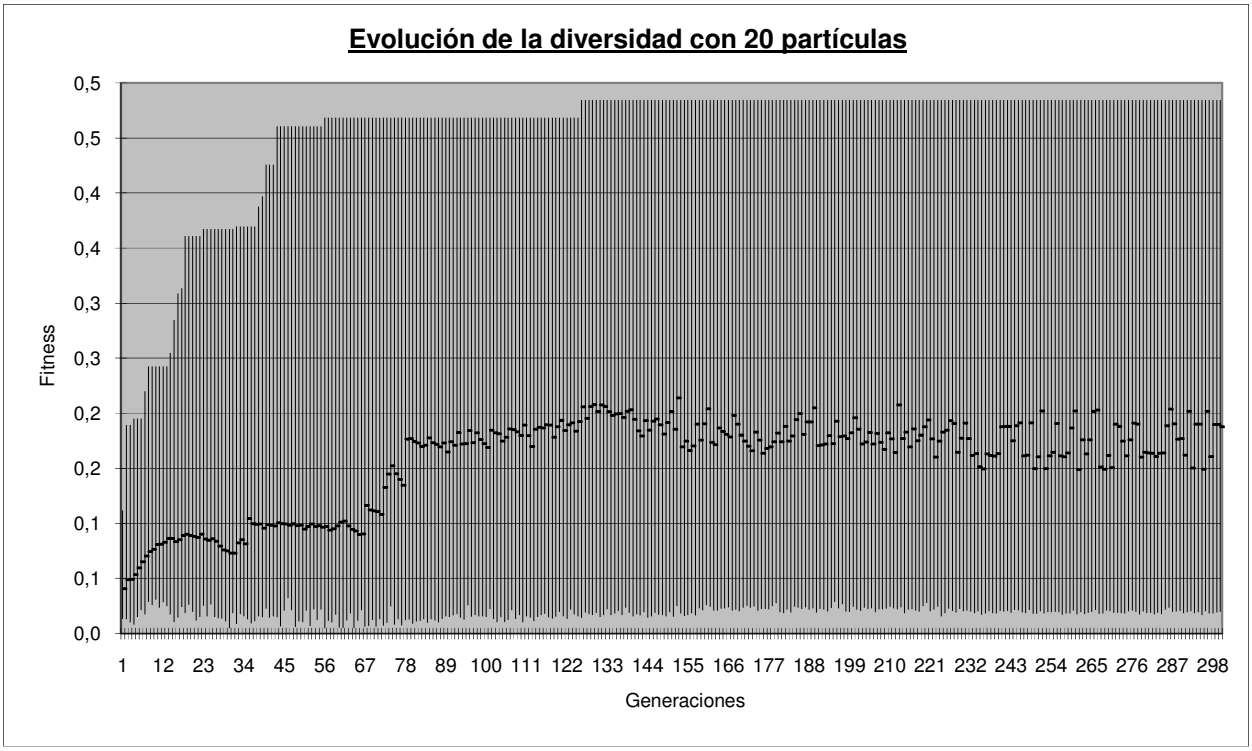


Figura 5

5. Conclusiones

Se ha presentado una estrategia basada en cúmulos de partículas de tamaño variable que permite entrenar redes neuronales. El mecanismo utilizado para incorporar nuevos individuos así como la forma de calcular el tiempo de vida preserva la diversidad de la población sin generar una cantidad excesiva de individuos.

Su aplicación en la resolución del juego de TaTeTi ha sido comparada con *gBest PSO* y *lBest PSO*.

Los resultados obtenidos muestran claramente que la estrategia propuesta en este artículo permite obtener redes con un mejor valor de aptitud que las producidas por ambas versiones de PSO con tamaño de población fijo. Es importante destacar que el algoritmo aquí propuesto genera redes neuronales con un valor de fitness equivalente a sus pares de población fija en una cantidad de generaciones muy inferior.

Aun resta evaluar el comportamiento de la estrategia propuesta en la resolución de algunas funciones complejas clásicas como las empleadas en [Lan00]. Actualmente se está aplicando este algoritmo en el aprendizaje de juegos de tablero más complejos.

6. Bibliografía

[Bir06] Bird S. and Li X. Adaptively Choosing Niching Parameters in a PSO. *Proceeding of Genetic and Evolutionary Computation Conference 2006 (GECCO'06)*, eds. M. Keijzer, et al., p.3 - 9, ACM Press. 2006.

[Lan00] Lanzarini L.,Sanz C. Naiouf M., Romero F. Mixed alternative in the assignment by classes vs. conventional methods for calculation of individuals lifetime in GAVaPS. *Proceedings of the 22nd International Conference on Information Technology Interfaces, 2000. ITI 2000*. pp. 383- 389. ISBN: 953-96769-1-6. June 2000

[Ken95] Kenedy J. and Eberhart R. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*. Vol IV, pp.1942-1948. Australia 1995

[Shi99] Shi Y., Eberhart R. An empirical study of particle swarm optimization. *Proceeding on IEEE Congress Evolutionary Computation*. pp.1945-1949. Washington DC, 1999.

[Shi98] Shi Y., Eberhart R. Parameter Selection in Particle Swarm Optimization. *Proceedings of the 7th International Conference on Evolutionary Programming*. pp. 591-600. Springer Verlag 1998. ISBN 3-540-64891-7

[Ber02] Van den Bergh F. An analysis of particle swarm optimizers. Ph.D. dissertation. Department Computer Science. University Pretoria. South Africa. 2002

[Cle02] Clerc M., Kennedy J. The particle swarm – explosion, stability and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*. Vol 6,nro. 1, pp. 58-73. Feb.2002

[Ert02] Ertöz L., Steinbach M. and Kumar V. A new shared nearest neighbor clustering algorithm and its applications. Proc. *Workshop on Clustering High Dimensional Data and its Applications*. Arlington, VA, USA, 2002, pp. 105--115.

[Fer05] Fernandes C., Ramos V., Rosa A. Varying the Population Size of Artificial Foraging Swarm on Time Varying Landscapes. Artificial Neural Networks: Biological Inspirations, Proc. ICANN´05: 15th Int. Conf., Warsaw, Poland, LNCS series, Vol. 3696, Part I, Springer-Verlag, pp. 311-316, 2005.

[Mei02] Meissner M., Schmuker M., Schneider G. Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural networks training. BMC Bioinformatics 2006; 7: 125. Published online 2006 March 10. DOI: 10.1186/1471-2105-7-125.

Global Numerical Optimization with a bi-population Particle Swarm Optimizer

Susana C. Esquivel and Leticia C. Cagnina
LIDIC (Research Group) - Universidad Nacional de San Luis
Ej. de Los Andes 950, (D5700HHW) San Luis, ARGENTINA
E-mails: {esquivel,lcagnina}@unsl.edu.ar

Abstract

This paper presents an enhanced Particle Swarm Optimizer approach, which is designed to solve numerical unconstrained optimization problems. The approach incorporates a dual population in an attempt to overcome the problem of premature convergence to local optima. The proposed algorithm is validated using standard test functions (unimodal, multi-modal, separable and nonseparable) taken from the specialized literature. The results are compared with values obtained by an algorithm representative of the state-of-the-art in the area. Our preliminary results indicate that our proposed approach is a competitive alternative to solve global optimization problems.

Keywords: Particle Swarm Optimizer, Optimization, Unconstrained Functions

Resumen

Este artículo presenta un nuevo algoritmo Particle Swarm Optimizer, diseñado para resolver problemas de optimización numéricos sin restricciones, que incorpora una población dual para intentar solucionar el problema de convergencia prematura en óptimos locales. El algoritmo propuesto es validado usando funciones de prueba estandard (unimodales, multi-modales, separables y no separables) tomadas de la literatura especializada. Los resultados son comparados con los valores obtenidos por un algoritmo representativo del estado del arte en el área. Los resultados preliminares indican que la propuesta es una alternativa competitiva para resolver problemas de optimización global.

Palabras claves: Particle Swarm Optimizer, Optimización, Funciones sin Restricciones

1 INTRODUCTION

In the last years metaheuristics (particularly, evolutionary algorithms) has been used to solve real-world applications. One of that metaheuristics that has been adopted to solve such problems is Particle Swarm Optimizer (PSO) [14].
PSO was conceived as a simulation of individual and social behavior [13] such as the one observed in flocks of birds and fish. PSO explores the search space using a population of individuals, and the best performers (either within a group or with respect to the entire population) affect the performance of the others. Each individual is named *particle* and represents a possible solution within a multidimensional search space. The particles have their own position and velocity, which are constantly updated. They record their past behavior and use it to move towards promising regions of the search space.
PSO has been found to be highly competitive for solving unconstrained real-world optimization problems [15, 16, 9, 3, 4]. However, on strongly multi-modal test functions, PSO tends to suffer premature convergence. This happens because the diversity decreases, leading to the stagnation of the swarm.

In this paper, we present a PSO algorithm which is designed to solve unconstrained optimization problems. Our approach contains a different mechanism to update the velocity and position of the particles [2], which is extended by adding to it a bi-population as a way to avoid premature convergence.

The remainder of this paper is organized as follows. Section 2 provides the statement of general unconstrained optimization problems. Section 3 briefly discusses the previous related work. In Section 4, we describe in detail our proposed approach. Section 5 describes the experimental setup and provides an analysis of the results obtained from our empirical study. Section 6 shows the statistical analysis of results between the algorithms presented. The conclusions and some directions for future research are stated in Section 7.

2 STATEMENT OF THE PROBLEM

The unconstrained optimization involves maximization and minimization problems, although maximization ones can be transformed into minimization ones as:

$$f'(\vec{x}) = -f(\vec{x}) \tag{1}$$

So, we treated just with minimization class of problem in this paper.
Without loss of generality, we can consider the general nonlinear optimization problem as a minimization problem, which can be formally stated as the problem of finding $\vec{x}$ which:

$$minf(\vec{x}) \text{ with } \vec{x} = (x_1, x_2, \ldots, x_D) \in \mathcal{S} \subseteq \mathcal{R}^D \tag{2}$$

Each $x_d \in [l_d, u_d]$ with $d \in [1..D]$. The l_d and u_d are the lower and upper bounds imposed on the decision variables. $\mathcal{S}$ (the search space) is a $D-$dimensional rectangle defined by the lower and upper bounds of each variable x_d.

3 PREVIOUS RELATED WORK

As indicated before, despite its success in a variety of optimization problems, PSO has been applied to solve just some unconstraint multi-modal function, but no much comparative studies with another

methods are presented. Next, we will review the most representative work done in this area.

Riget et al. [19] presented an *attractive* and *repulsive* PSO (ARPSO) specially designed to overcome the problem of premature convergence. The algorithm uses a measure to control the diversity in the swarm. The performance of ARPSO was compared with a basic PSO and a genetic algorithm using 4 multi-modal functions. The conclusions are that their algorithm performs extremely well for the problems tested.

In [10], Jian discusses the parameters on PSO. A velocity and position disturbance mechanism is introduced to prevent premature convergence. The algorithm is tested using 2 unimodal function and 2 multi-modal. The simulation experiments showed that the improved PSO has good performance.

Voss [20] introduced a PCPSO procedure to create a *flying and dynamic* coordinate system with the particles. The author compared the methodology of the algorithm as a symbiotic relationship between swarm movement and the rotation and dimension of the life space. The proposed PCPSO is validated with 4 function: 2 unimodal and 2 multi-modal. The algorithm performed well for the 30-dimensional test functions tested.

Oltean et al. [5] evolved the structure of a Particle Swarm Optimization with a new model. The model is a hybrid method that combines a basic PSO and a genetic algorithm. The genetic algorithm chromosomes are vectors with the indexes of the particles. Those chromosomes are used to select which particles will be update. The authors make some experiments with 6 unimodal and 4 multi-modal functions. The results are compared with the previous version PSO standard and the evolved PSO. The conclusions are that sometimes the new version performs similarly and sometimes better.

Liang et al. [17] showed a variant of PSO which uses a novel learning strategy whereby all other particles' best is used to update the velocity of particles in the swarm. The strategy helps to preserve diversity and in that way avoid premature convergence. The paper studies the algorithm performance using 2 unimodal functions, 6 multi-modal unrotated, 2 noncontinuos Rastrigin's function and 6 rotated multi-modal problems. The results showed that the proposed algorithm improves significantly the performance of PSO algorithms.

In [7] the authors proposed the use of two simple PSO. Both (global and local models) were hybridized with a nonuniform mutation operator taken from the evolutionary algorithms literature. The algorithms were tested with 4 multi-modal functions and compared the results with those obtained with 6 PSO models studied by Peer et al. The performance of their algorithms indicate that the proposed PSO models are highly competitive.

4 BI-PSO ALGORITHM

In this section, we describe in detail our proposed approach, which we call the Bi-PSO algorithm.

4.1 General Model

As stated before, a PSO algorithm operates on a population of particles. Due to the type of problem to optimize (with n decision variables), the particles are n-dimensional real number vectors. The best position found so far for the particles (for the *gbest* model) or in the neighborhood (*lbest* model) is

recorded. The best value reached by each particle (*pbest*) is stored, too. As in the basic model, the particles evolve using two update formulas, one for position and another one for velocity.

4.2 Our Approach

As it was stated in some of our previous work [1], the *gbest* model tends to converge to a local optimum although works well in many problems. Motivated by this, we proposed a formula to update the velocity, using a combination of both the *gbest* and the *lbest* models [2], it is shown in equation (3).

$$v_{id} = w(v_{id} + \gamma_1(p_{id} - par_{id}) + \gamma_2(p_{ld} - par_{id}) + \gamma_3(p_{gd} - par_{id})) \tag{3}$$

where v_{id} is the velocity of particle i at the dimension d; w is the inertia factor [6] whose goal is to balance global exploration and local exploitation. γ_1 is the personal learning factor, and γ_2 and γ_3 are the social learning factors. These 3 values are multiplied by 3 different random numbers within the range [0..1], p_{id} is the best position reached by the particle i; p_{ld} is the best position reached by any particle in the neighborhood, p_{gd} is the best position reached by any particle in the swarm. par_{id} is the value of the particle i at the dimension d.

The equation for updating the particles also was modified as we proposed in a previous work [2]. In that paper, during 10% of the iterations, we applied the normal formula (depicted in equation (4)) as suggested in [14].

$$par_{id} = par_{id} + v_{id} \tag{4}$$

And in the remainder 90% of cases, we used equation (5) proposed by Kennedy [12].

$$par_i = N\left(\frac{p_i + p_l}{2}, |p_i - p_l|\right) \tag{5}$$

where p_i is the position of the particle to be updated, N is the Gaussian random generator, p_i and p_l are the best position reached by the particle par_i and the best position reached by any particle in the neighborhood of par_i, respectively. That probability was empirically found to be the best after performing a series of experiments with all the test functions evaluated.

We use a circle topology [11] to compute the p_{ld} value, in which each particle is connected to k neighbors. The neighbors are determined by the position of the particles in the storage structure. Figure 1 illustrates this concept.

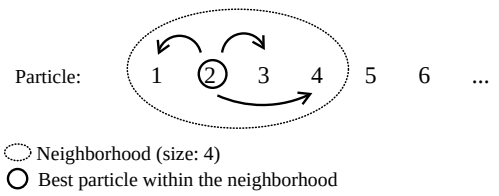


Figure 1: Circle topology adopted in the population of our PSO approach

To help avoiding convergence to a local optimum, we used a dynamic mutation operator [1] which is applied to each individual with a pm-probability. This value is calculated considering the total number of iterations in the algorithm (*cycles*) and the current cycle number as the following equation indicates:

$$pm = max\_pm - \frac{max\_pm - min\_pm}{max\_cycle} * current\_cycle \quad (6)$$

where max_pm and min_pm are the maximum and minimum values that pm can take, max_cycle is the total number of cycles that the algorithm will iterate, and $current\_cycle$ is the current cycle in the iterative process. The probability pm is high in the early cycles and we decrease that value on every iteration. That helps to explore at the starts in the search process and to explode at the ends.

4.3 Bi-population

The Bi-PSO algorithm splits the entire population into two subpopulations each of which is independently evolved. The idea is to maintain more than one group of particles exploring the search space (at the same time). In that way the possibility of falling into local optima is reduced.

One may then wonder why to adopt only two subpopulations and not more. The reason is that it does not make any sense to adopt more than two subpopulations, considering the small number of particles that we use in our original population (only 10). In fact, we believe that our neighborhood topology would not work properly if we adopt less than 5 particles and therefore our choice of adopting only two subpopulations.

All the features stated before for the entire population (neighborhoods, *lbest* and *gbest* approaches, equations for updating the velocity and the positions) still apply, but in this case, they are applied not to a single population, but to each subpopulation. When the iterative process finishes, the best particle from both subpopulations is reported as the final output.

4.4 Bi-PSO Pseudocode

Figure 2 shows the pseudocode of our proposed Bi-PSO algorithm. At the beginning of the search, we initialize the vectors of position and velocity of each particle in both subpopulations (lines 2 to 5). After evaluating the particles and obtaining the best values: *pbest*, *lbest* and *gbest* (lines 6 and 7), the subpopulations begin to evolve. During the evolutionary process, new values of *pbest*, *lbest* and *gbest* are chosen and both, the velocity and the position of each particle are updated (lines 8 to 24). At line 25, a *keeping* mechanism is applied to control that all the dimensions in all particles are within the allowable bounds. The mutation probability is updated and the particles are mutated, if applicable (lines 26 and 27). After that, the particles are evaluated and new "best" values are recorded (lines 28 to 30). Finally, the best value reached by any subpopulation is taken and compared. The best of them is returned (lines 31 and 32).

5 EXPERIMENTAL STUDY

For validating our proposed approach, we adopted 13 minimization test problems with 30 variables taken from [21]:

- f_1: **Sphere Model**. Unimodal and separable function.
- f_2: **Schwefel's Problem 2.22**. Unimodal and separable function.
- f_3: **Schwefel's Problem 1.2**. Unimodal and nonseparable.
- f_4: **Schwefel's Problem 2.21**. Unimodal and separable function.

```
0.  Bi-PSO:
1.  Swarm Initialization
2.    Initialize subpop1
3.    Initialize velocity for subpop1
4.    Initialize subpop2
5.    Initialize velocity for subpop2
6.    Evaluate fitness for each subpop
7.    Record pbest and gbest for each subpop
8.  Swarm flights through the search space
9.    DO
10.     FOR each subpop DO
11.         FOR i=1 TO numberOfparticles DO
12.             Search the best leader in the
13.                neighborhood of part_i
14.                and record in lbest_i
15.             FOR j=1 TO numberOfdimensions DO
16.                 Update vel_ij
17.                 IF flip(0.1)
18.                    Update part_ij with eq.(3)
19.                 ELSE
20.                    Gaussian update with eq.(5)
21.                 END
22.             END
23.         END
24.     END
25.     Keeping particles
26.     Update pm
27.     Mutate every particle depending on pm
28.     Evaluate fitness(part_i)
29.     Record pbest and gbest
30.   WHILE(current_cycle < max_cycle)
31.   result=BEST(best_subpop1,best_subpop2)
32.   RETURN(result)
```

Figure 2: Pseudo-code of Bi-PSO

- f_5: **Generalized Rosenbrock's Function**. Multi-modal and nonseparable problem.
- f_6: **Step Function**. Unimodal and separable problem.
- f_7: **Quartic Function with Noise**. Unimodal and separable problem.
- f_8: **Generalized Schwefel's Problem 2.26**. Multi-modal and separable function, with many local minima. The number of local minima increases exponentially as the function dimension increases.
- f_9: **Generalized Rastrigin's Function**. Multi-modal and separable problem, with many local minima. The number of local minima increases exponentially as the function dimension increases.
- f_{10}: **Ackley's Function**. Multi-modal and nonseparable problem, with many local minima. The number of local minima increases exponentially as the function dimension increases.
- f_{11}: **Generalized Griewank Function**. Multi-modal and nonseparable problem, with many local minima. The number of local minima increases exponentially as the function dimension increases.

- f_{12} and f_{13}: **Generalized Penalized Functions**. Multi-modal and nonseparable problems, with many local minima. The number of local minima increases exponentially as the function dimension increases.

The detailed description of the test problems may be consulted in [21]. We performed 50 independent runs per problem, with a total of 120,000 evaluations of objective function per run. Our proposed Bi-PSO used the following parameters: swarm size = 10 particles, $pm\_min = 0.1$, $pm\_max = 0.4$, neighborhood size = 3, inertia factor w = 0.8, personal learning factor and social learning factors for γ_1, γ_2 and γ_3 was set to 1.8. The parameter settings such as swarm size, mutation probability, neighborhood size and learning factors were empirically derived after numerous experiments.

Our results were compared with respect to those obtained by a Differential Evolution (DE) algorithm proposed by Velázquez [18]. DE is an Evolutionary Algorithm created to solve optimization problems, mainly in continuous search spaces. It is well known the high performance of DE in the most problems tested, compared with other optimization techniques. DE performs mutation based on the distribution of solutions in the population of individuals, so search directions depend on individuals' location selected to calculate the mutation values.
In [18] a new approach is presented to increase the probability of each parent to generate a better offspring. For that the author allows each solution to generate more than one offspring using a different operator which combines the information of the best solution and of the current parent to find new search directions. The new model uses a binomial discrete recombination operator, too. The parameter setting may be consulted at [18], but it is important remark that the DE performed 50 independent runs with a total of 120.000 evaluations (the same that with Bi-PSO) as the author suggest in his work. DE approach is highly competitive and is representative of the state-of-the-art in the area, for that we use it to make the comparison of performance.
Table 1 shows the best values found by Bi-PSO and DE.

Table 1: Comparison of the **best** values obtained by our Bi-PSO and the Differential Evolution (DE).

Funct.	Benchmark	Bi-PSO	DE
f1	0.00000	0.00000	0.00000
f2	0.00000	0.00000	0.00000
f3	0.00000	3.50228	**0.00000**
f4	0.00000	0.11688	**0.00000**
f5	0.00000	0.**00010**	0.04031
f6	0.00000	0.00000	0.00000
f7	0.00000	0.00000	0.00000
f8	-12569.48661	-12569.48661	-12569.48661
f9	0.00000	0.00000	0.00000
f10	0.00000	0.00000	0.00000
f11	0.00000	0.00000	0.00000
f12	0.00000	0.00000	0.00000
f13	0.00000	0.00000	0.00000

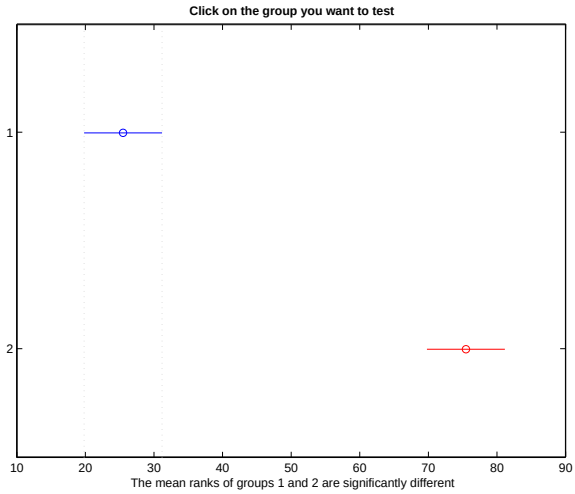
Table 2 shows the Mean values obtained by Bi-PSO and DE for the 13 unconstrained test functions adopted in our empirical study.

Table 2: Best Values obtained with **Bi-PSO** and **DE**.

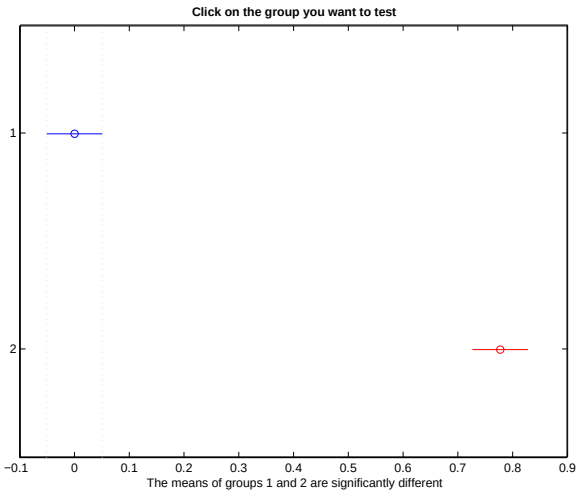
Funct.	Benchmark	Mean DE	Mean Bi-PSO
f1	0.00000	0.00000	0.00000
f2	0.00000	0.00000	0.00000
f3	0.00000	**0.00000**	178.54809
f4	0.00000	**0.00000**	0.77809
f5	0.00000	**2.71842**	28.28478
f6	0.00000	0.00000	0.00000
f7	0.00000	0.00000	0.00000
f8	-12569.48661	-12550.53648	**-12569.48632**
f9	0.00000	**0.23879**	0.84162
f10	0.00000	0.00000	0.00000
f11	0.00000	0.00000	0.00000
f12	0.00000	0.00000	0.00000
f13	0.00000	0.00000	0.00000

6 STATISTICAL ANALYSIS

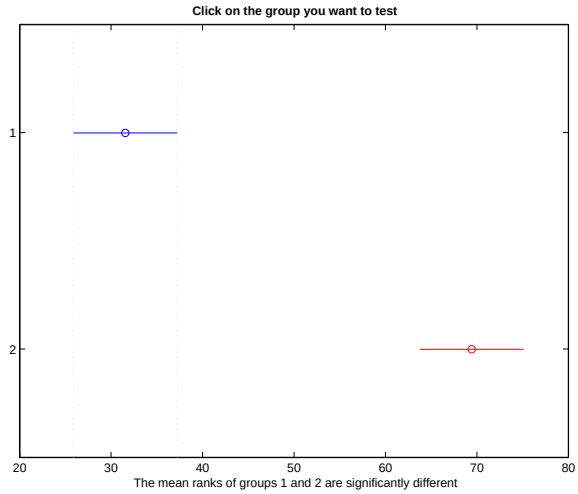
To analize the performance of our algorithm we used a statistical test. We do an analysis of variance between DE and Bi-PSO using the best values of the 50 independent runs we did with each one. We apply the Kruskal-Wallis [8] nonparametric one-way analysis because the values (the sample) do not have a normal distribution (determined with the Kolmogorov-Smirnov test).
The Kruskal-Wallis test returns the *p-value* for the null hypothesis for all samples. If the *p-value* is zero or near, that suggests that at least one sample is significantly different (or *statistically significant*) than the other samples. Usually, if *p-value* is less than 0.05, we declare the results are significant.
Table 3 shows the *p-value* for each function. The results indicate the values reached with Bi-PSO (for all function except: f3, f4, f5, f8 and f9), are not statistically significant from those of DE. That indicates DE and Bi-PSO are comparable with respect a performance. As we know, DE is a higher performance approach even its complex process to compute the solutions. Bi-PSO is simpler and has the similar performance in 8 functions.
For f3, f4, f5, f8 and f9, the values reached by DE are statistically different (table 3) from those obtained with our algorithm. For functions f3 and f4 DE obtains the best results as we can observe in table 1. However, for function f5 the best value is reached by our algorithm (table 1). For functions f8 and f9, we do not observe differences between the bests obtained by every algorithm (table 1), but we observe the differences in table 2 in which the best mean value for f8 is obtained for Bi-PSO and for function f9 is obtained by DE.
For those functions significantly different we apply the Tukey test to determine between which experimental conditions the differences are significant. The test returns: an estimate value (EV) into a range ([LI,LS]). If the range does not contain the zero-value, then the results are confirmed (significantly different). As we observe in table 3, for all function statistically significant the ranges do not have the zero-value. The subfigures in figure 3 confirm the last (algorithm 1 is DE and algorithm 2 is Bi-PSO, on the vertical axis).



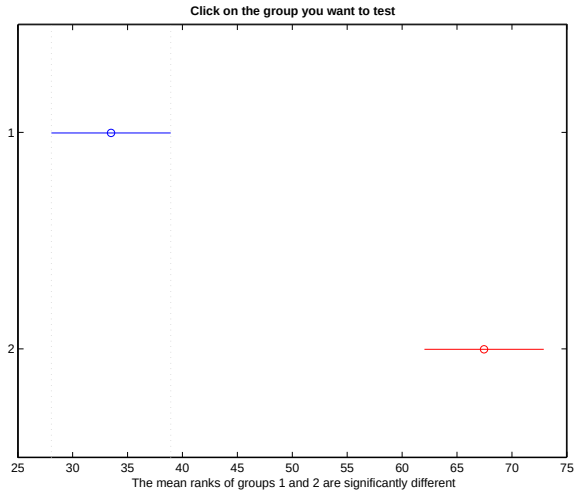
(a) f3



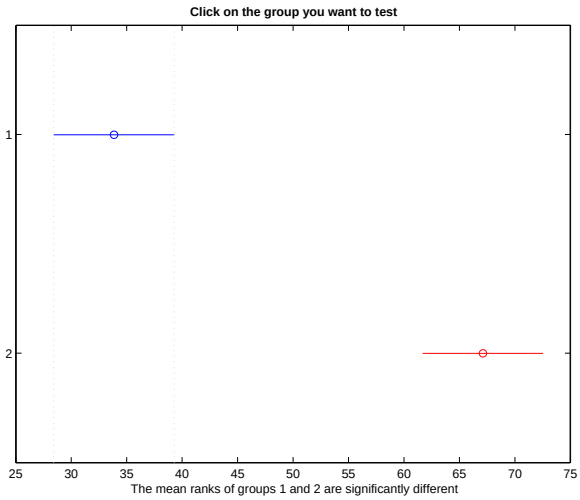
(b) f4



(c) f5



(d) f8



(e) f9

Figure 3: DE vs Bi-PSO Tukey's test

Table 3: Kruskal-Wallis' p-values y Tukey's results for DE vs Bi-PSO

Funct.	*p-value*	LI	EV	LS
f1	1.0000	-	-	-
f2	1.0000	-	-	-
f3	0.0000	-61.3721	-50.0000	-38.6279
f4	0.0000	-0.8797	-0.7780	-0.6763
f5	6.6437e-011	-49.2521	-37.8800	-26.5079
f6	1.0000	-	-	-
f7	1.0000	-	-	-
f8	8.9338e-010	-44.8755	-34.0000	-23.1245
f9	2.0054e-009	-44.1559	-33.2800	-22.4041
f10	1.0000	-	-	-
f11	0.3162	-	-	-
f12	1.0000	-	-	-
f13	1.0000	-	-	-

7 CONCLUSIONS

We have introduced a bi-population PSO algorithm, which is proposed to solve unconstrained numerical optimization. The results reached by Bi-PSO are good and competitive with respect to Differential Evolution (DE) algorithm (which is one of the best evolutionary algorithms to treat constrained and unconstrained problems). Our results indicate that Bi-PSO, which is simpler than DE obtains comparable performance for the best values found although it presented more variability in the mean values. As part of our future work, we aim to study alternative schemes to maintain diversity. Another goal is to improve the robustness of our approach, so that the variability of results significantly decreases, without degrading the quality of the best solutions found.

8 ACKNOWLEDGEMENTS

The authors would like to thank Mg. Jesús Velázquez Reyes and PhD. Carlos Coello Coello to provide us the Differential Evolution algorithm source code in order to make the proofs. Also we gratefully acknowledge constant support from the University and ANPCYT.

REFERENCES

[1] L. Cagnina, S. Esquivel, and R. Gallard. Particle swarm optimization for sequencing problems: a case study. In *Congress on Evolutionary Computation*, pages 536–541, Portland, Oregon, USA, 2004. http://www.lidic.unsl.edu.ar/publicaciones/in- fo_publicacion.php?id_publicacion=200.

[2] L. C. Cagnina, S. C. Esquivel, and C. A. Coello Coello. A particle swarm optimizer for constrained numerical optimization. In *9th International Conference - Parallel problem Solving from Nature - PPSN IX*, pages 910–919, Reykjavik, Island, 2006.

[3] W. Cedeno and D. Agrafiotis. Particle swarms for drug design. In *Proceeding of the IEEE Congress on Evolutionary Computation 2005*, pages 479–486, Edinburgh, Scotland, 2005.

[4] E. Correa, A. Freitas, and C. Johnson. A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set. In *Proceeding of the GECCO 2006*, pages 35–42, Seattle, Washington,USA, 2006.

[5] L. Diosan and M. Oltean. Evolving the structure of the particle swarm optimization algorithms. In J. Gottlieb and G. R. Raidl, editors, *EvoCOP 2006, LNCS 3906*, pages 25–36. Springer-Verlag Berlin Heidelberg 2006, 2006.

[6] R. Eberhart and Y. Shi. A modified particle swarm optimizer. In *International Conference on Evolutionary Computation, IEEE Service Center*, Anchorage, AK, Piscataway, NJ, 1998.

[7] S. Esquivel and C. Coello Coello. On the use of particle swarm optimization with multimodal functions. In IEEE Press, editor, *Proceedings of the Congress on Evolutionary Computation (CEC 2003)*, Canberra, Australia, December 2003.

[8] M. Hollander and D. A. Wolfe. *Nonparametric Statistical Methods*. Wiley, 1973.

[9] X. Hu, R. Eberhart, and Y. Shi. Swarm intelligence for permutation optimization: a case study on n-queens problem. In *Proceeding of the IEEE Swarm Intelligence Symposium*, pages 243–246, Indianapolis, Indiana, USA, 2003.

[10] W. Jian, Y. Xue, and J. Qian. An improved particle swarm optimization algorithm with disturbance. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, pages 5900–5904, 2004.

[11] J. Kennedy. Small world and mega-minds: effects of neighborhood topologies on particle swarm performance. In *1999 Congress on Evolutionary Computation*, pages 1931–1938, Piscataway, NJ, 1999. IEEE Service Center.

[12] J. Kennedy. Bores bones particle swarm. In *IEEE Swarm Intelligence Symposium*, pages 80–87, 2003.

[13] J. Kennedy and R. Eberhart. The particle swarm: social adaptation in information-processing systems. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Organization*, 1999.

[14] J. Kennedy and R. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, California, USA, 2001.

[15] J. Kennedy and W. Spears. Matching algorithm to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In *IEEE Conference on Evolutionary Computation*, Anchorage, Alaska, 1998.

[16] R. Krohling, H. Knidel, and Y. Shi. Solving numerical equations of hydraulic problems using particle swarm optimization. In *Congress on Computational Intelligence*, pages 1968–1961, Honolulu, Haway, 2002.

[17] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar. Comprehensive learning particle sawrm optimizer for global optimization of multimodal functions. In *IEEE Transactions on Evolutionary Computation*, volume 10, pages 281–295, June 2006.

[18] Jesús Velázquez Reyes. Propuesta de evolución diferencial para optimización de espacios restringidos. Master's thesis, Departamento de Ingeniería Eléctrica, Sección de Computación, Centro de Investigación y de Estudios Avanzados del IPN, México, 2006.

[19] J. Riget and S. Vesterstrom. A diversity-guided particle swarm optimizer - the arpso. Technical report, EVALife, 2002. http://www.evalife.dk.

[20] M. Voss. Principal component particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 2005.

[21] X. Yao and Y. Liu. Fast evolution strategies. In T. Bäck G. Rozenberg and A. E. Eiben, editors, *Advances in Evolutionary Computing: theory and applications*, pages 45–94, New York, USA, 2003. Springer-Verlag New York.

Agent Programming using Defeasible Argumentation for Knowledge Representation and Reasoning

Sebastián Gottifredi Alejandro J. García Guillermo R. Simari
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET),
Laboratorio de Investigación y Desarrollo de Inteligencia Artificial,
Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur,
Avenida Alem 1253,(B8000BCP), Bahía Blanca, Argentina
Tel: (0291) 459-5135 / Fax: (0291) 459-5136
Email: {sg,ajg,grs}@cs.uns.edu.ar

Abstract

In this work two declarative approaches based on the BDI theory are studied, an agent programming language 3APL and an agent architecture that uses defeasible argumentation for knowledge representation and reasoning. Based on that study and considering that in 3APL the knowledge representation language is not fixed, we will propose 3APL-DeLP where the agent knowledge is represented by a DeLP-program and the agent may reason with a defeasible argumentation formalism.

Keywords: Agent Programming Languages, Agent Architectures, Agent Theories, Defeasible Argumentation.

1 INTRODUCTION

In this work two declarative approaches based on the BDI theory are studied. The 3APL programming language, was chosen because it allows a declarative specification and implementation of the agents, where agents can communicate and can reach their goals using plans. The agent architecture proposed in [7],[6] (BDI-DeLP), was chosen because it uses Defeasible argumentation for the knowledge representation and reasoning of the agents. Based on the study of the approaches and considering that in 3APL the knowledge representation language is not fixed, we will propose 3APL-DeLP where the agent knowledge is represented by a DeLP-program and the agent may reason with a defeasible argumentation formalism.

The importance of using intelligent agents based on mental components like Beliefs, Desires, Commitments and Intentions to solve complex problems is well known in the literature [11], especially those agents based on BDI theory [1]. Nowadays, tools are needed to specify and programs agents in terms of these components. Several programming Languages and architectures based on BDI are proposed in the literature. However, only some of them allow to specify agents in a declarative way. In particular, we are interested in agent development tools that provide an argumentative mechanism for agent reasoning, besides a declarative way to specify its mental components.

This work is organized as follows. In Section 2 an introduction to 3APL and BDI-DeLP approaches is made, indicating goals, advantages and scope for each of them. In Section 3 examples for the approaches are shown. Then an analysis of how the mental components are specified and how reasoning process works for each approach is made. In Section 4 3APL-DeLP is specified. Finally, in Section 5 we expose some conclusions and areas of future research.

Partially supported by CONICET (PIP 5050), Universidad Nacional del Sur and Agencia Nacional de Promoción Científica y Tecnológica.

2 TWO APPROACHES BASED ON THE BDI THEORY

In this section, two approaches that use BDI-Theory [1], are introduced. As exposed in [11] the research on agent programming can be divided in three areas: agent theories, agent architectures and agent programming languages. Agent theories are related to the formalisms that specify the properties of the agents. Agent architectures are related to the building of mental components and their interactions, satisfying the properties of an agent theory. Finally, agent programming languages are related to the primitives used to program and implement agents specified by agent theory.

This work is focused on the BDI agent theory. This theory exposes that the mental state of an agent is composed by three components which try to capture the intuitive meaning of Beliefs, Desires and Intentions. The BDI theory is well known in the literature and several agent architectures and Languages have been defined using it. In this work, two of these approaches will be considered: an agent architecture called BDI-DeLP, and an agent programming language called 3APL.

2.1 A BDI agent architecture with a Dialectical Framework: BDI-DeLP

In [7] and [6] a BDI agent architecture based on a Dialectical Framework was defined. The goal of this proposal is to allow the specification of the agent mental components and their relations, using Defeasible Logic Programming (DeLP) [4]. This architecture is based on the BDI theory. It introduces a formalism to allow the specification of Beliefs, Desires and Intentions using DeLP programs. Thus, the agent may reason with a defeasible argumentation formalism. Therefore this proposal will be referred as BDI-DeLP in the rest of this work.

BDI-DeLP has advantages as an agent architecture: BDI agents mental components and their interaction can be specified in a declarative way (see Fig.1), and it uses a high level mechanism (DeLP) for knowledge representation and reasoning.

Next, we give a brief summary of DeLP (for more details see [4]). The goal of this summary is to introduce the concepts of DeLP that will be used in the analysis of further sections. In DeLP, knowledge is represented using facts, strict rules, and defeasible rules:

- *Facts* are ground literals representing atomic information or the negation of atomic information using strong negation "$\sim$".
- *Strict Rules* are denoted $L_0 \leftarrow L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground extended literals ($e.g.\ \sim a \leftarrow b$).
- *Defeasible Rules* are denoted $L_0 \prec\!\!- L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground extended literals. ($e.g.\ c \prec\!\!- \sim d \quad f \prec\!\!- not\ g$).

Rules are distinguished by the type of arrows, and a defeasible rule "$Head \prec\!\!- Body$" expresses that "*reasons to believe in the antecedent* $Body$ *give reasons to believe in the consequent* $Head$" representing tentative information that may be used if nothing could be posed against it.

A Defeasible Logic Program (DeLP) $\mathcal{P}$ is a set of facts, strict rules and defeasible rules. When required, $\mathcal{P}$ is denoted (Π, Δ) distinguishing the subset Π of facts and strict rules, and the subset Δ of defeasible rules. Strict and defeasible rules are ground, however, following the usual convention, some examples will use "schematic rules" with variables.

Strong negation could appear in the head of program rules, and can be used to represent contradictory knowledge. From a program (Π, Δ) contradictory literals could be derived. however, the set Π (used to represent non-defeasible information) must be non-contradictory, $i.e.$no pair of contradictory literals can be derived from Π. Given a literal L, $\overline{L}$ represents the complement with respect to strong negation.

If contradictory literals are derived from (Π, Δ), a dialectical process is used for deciding which literal prevails. In short, an *argument* for a literal L, denoted $\langle \mathcal{A}, L \rangle$, is a minimal set of defeasible

rules $\mathcal{A} \subseteq \Pi$, such that $\mathcal{A} \cup \Pi$ is non-contradictory, and there is a derivation for L from $\mathcal{A} \cup \Pi$. A literal L is *warranted* from (Π, Δ) if there exists a non-defeated argument $\mathcal{A}$ supporting L. To establish if $\langle \mathcal{A}, L \rangle$ is a non-defeated argument, *argument rebuttals* or *counter-arguments* that could be *defeaters* for $\langle \mathcal{A}, L \rangle$ are considered, $i.e.$counter-arguments that by some criterion are preferred to $\langle \mathcal{A}, L \rangle$. A defeater $\mathcal{A}_1$ for an argument $\mathcal{A}_2$ can be proper ($\mathcal{A}_1$ *stronger than* $\mathcal{A}_2$) or *blocking* (same strength). In the examples of this paper we assume generalized specificity as the comparison criterion, however, as explained in [4] the criterion could be easily changed.

Since defeaters are arguments, there may exist defeaters for them, and defeaters for these defeaters, and so on. Thus, a sequence of arguments called *argumentation line* is constructed, where each argument defeats its predecessor in the line (for a detailed explanation of this dialectical process see [4]). In DeLP, a query Q could have four possible answers: YES, if Q is warranted; NO, if the complement of Q is warranted; UNDECIDED, if neither Q nor its complement is warranted; and UNKNOWN, if Q is not in the signature of the program.

2.2 An agent programming Language: 3APL

In contrast with the BDI-DeLP that provides an agent architecture, 3APL [5] provides an agent programming language. 3APL was presented in [5] and extended in [3] and follows the spirit of Shoham's Agent Oriented Programming paradigm [8]. Its goal is to allow the programmer the creation of cognitive agents that can interact each other. As an agent programming language 3APL [11] is based on the BDI agent theory, and provides a concrete language to program and execute agents. The agents of 3APL combine logic programming (for the specification of their mental components) and imperative programming (for the structure of their plans).

3APL has advantages as an agent programming language: Agents can be specified and implemented in a declarative way through their mental components (see Fig.2), these mental components follow the BDI mental components, agents can solve their goals via plans and communication between agents can be specified and affects the agent mental components. The authors of 3APL have developed a platform[9] were agents can executed and tested.

3 AN ANALYSIS OF 3APL AND BDI-DELP

In this section an analysis of 3APL and BDI-DeLP will be made. The goal of this analysis is to expose strong and weak points of each approach, and show how these alternatives can complement each other.

This analysis will be centered in knowledge representation and Reasoning. Therefore, the analysis will be divided in two categories: Mental Components and Reasoning mechanism where the mental component area will be divided in three subcategories: Beliefs, Desires and Intentions.

Since 3APL is agent programming language, it provides support for full agent implementation. Thus, communication and planning, will not be analysed in deep in this work because these items are only supported by 3APL. The communication model used in 3APL is presented in [10] and is based on the *FIPA* standards. This model is integrated with mental components of the agent. Plans in 3APL can be a basic action or a structure of basic actions. This structure follows the spirit of an imperative programming language.

Now we will introduce a working example in order to show and analyze the main differences between both approaches.

Example 1 *Consider a cognitive agent with the following features:*

- *it has five signal sensors for perceiving the environment. If s_i is present in the agent beliefs means that the sensor i is active.*
- *it can infer new beliefs from active sensors: b_1, if s_1 or s_2 are active; b_2 if s_3 is active; $\sim b_1$, if s_1 and s_2 are both active, or if s_5 is active; and $\sim b_2$, if s_2 and s_4 are both active.*
- *it has two desires: g_1, achievable if it believes in b_1 and not achievable if it believes in b_2; and g_2, achievable if s_3 is active and not achievable if it believe in b_1.*
- *it has: two ways to achieve the desire g_1, one when s_1 and s_4 are both active, and the other only s_4 is active; and one way to achieve the desire g_2, that is when s_4 is active.*

The agent of Ex.1 has not necessarily a real connotation, it is a brief example that will be used just to show the representational capabilities of both approaches. Next, we will show a specification for the agent of Ex.1 in BDI-DeLP (Fig.1) and in 3APL (Fig.2). The goal of these figures is to show similarities and differences between both approaches. This specification will represent a snapshot of the agent mental state in an arbitrary moment, where the sensors s_1, s_3, s_4, s_5 are active. The mental components of the agent of each figure will be detailed below.

The specification of the agent of Ex.1 with BDI-DeLP is shown below in Fig.1. In this specification mental components like, Beliefs (Φ and BR), Desires (D), filtering rules for the desires ($\mathcal{P}_{\mathcal{F}}$) and the intention rules (IR), can be differentiated. There, Φ has facts used to represents sensors, BR has rules used to obtain inferred beliefs, D has facts used to represents the desires, $\mathcal{P}_{\mathcal{F}}$ has rules used to determine if a desire is achievable, and IR has rules used to determine which desire the agent will try to achieve next.

$$\Phi = \left\{ \begin{array}{l} s_1 \\ s_3 \\ s_4 \\ s_5 \end{array} \right\} \quad BR = \left\{ \begin{array}{l} b_1 \prec s_1 \\ b_1 \prec s_2 \\ \sim b_1 \prec s_5 \\ \sim b_1 \prec s_1, s_2 \\ b_2 \prec s_3 \\ \sim b_2 \prec s_2, s_4 \end{array} \right\} \quad D = \left\{ \begin{array}{l} g_1 \\ g_2 \end{array} \right\}$$

$$\mathcal{P}_{\mathcal{F}} = \left\{ \begin{array}{l} g_1 \prec b_1 \\ \sim g_1 \prec b_2 \\ g_2 \prec s_3 \\ \sim g_2 \prec \sim b_1 \end{array} \right\} \quad IR = \left\{ \begin{array}{l} g_1 \Leftarrow \{s_1, s_4\}\{\} \\ g_1 \Leftarrow \{s_4\}\{\} \\ g_2 \Leftarrow \{s_4\}\{\} \end{array} \right\}$$

Figure 1: BDI-DeLP Agent example

The specification of the agent of Ex.1 with 3APL is shown below in Fig.2. In this specification, Beliefs (`BELIEF BASE`), capabilities to manipulate the beliefs (`CAPABILITIES`), Desires (`GOAL BASE`) and the reasoning rules (`RULEBASE`), can be differentiated. There, `BELIEF BASE` has facts and rules used to represent the sensors and the inferred beliefs, `CAPABILITIES` has rules used to represent updates of the sensors, `GOAL BASE` has facts used to represent the desires, and `RULEBASE` has rules used to select a desire and a plan for achieving it.

In Fig.2 in order to obtain an equivalent version of the agent specified in Fig.1, the defeasible rules and the argumentation process were emulated. This will be described in detail in the rest of the section.

```
CAPABILITIES {
    {perception(Y)} Add(X) {not perception(Y), perception(X)}
} BELIEFBASE {
   perception([s1, s3, s4, s5]),
   perceived(X):-perception(L),member(X,L).
   b1:- perceived(s1), not tail1, not perceived(s5).
   b1:- perceived(s2), not tail1, not perceived(s5).
   tail1:- perceived(s1), perceived(s2).
   b2:- perceived(s3), not tail2.
   tail2:- perceived(s2), perceived(s4).
} GOALBASE {
    g1, g2
} RULEBASE {
    g1 <- perceived(s1), perceived(s4), b1, not(b2) | {Plan1Goal1, Sense(X), Add(X)},
    g1 <- perceived(s4), b1, not(b2) | {Plan2Goal1, Sense(X), Add(X)},
    g2 <- perceived(s4), perceived(s3), not(tail1) | {PlanGoal2, Sense(X), Add(X)},}
```

Figure 2: 3APL Agent example

3.1 Agent mental components

As stated above, both 3APL and BDI-Delp are based on the BDI agent theory. Agents specified using the BDI theory are described by three mental components: Beliefs, Desires and Intentions. Thus the analysis in this section will be divided in three parts, one for each mental component of the BDI theory.

3.1.1 Beliefs

In the BDI-DeLP proposal, agent beliefs are specified by a DeLP program called $\mathcal{P}_B = (\Pi_B, \Delta_B)$. In Π_B two disjoint subsets will be distinguished: Φ of perceived beliefs that will be updated dynamically, as the agent perceives the environment, and Σ of strict rules and facts that represent static knowledge. Thus $\Pi_B = \Phi \cup \Sigma$. The agent of Fig.1 has $\Sigma = \emptyset$. Besides the perceived beliefs, the agent may use strict and defeasible rules from $\mathcal{P}_B$ to obtain a warrant for its derived beliefs. The facts of Σ are not perceived, they do represent agent features, roles, etc. Therefore an agent in this proposal will have different types of beliefs:

- ***Perceived belief***:it is a fact of Φ that the agent has perceived directly from its environment.
- ***Strict belief***: it is a literal that is not a perceived belief, and its derived only from Π_B.
- ***Defeasible belief***: it is a warranted literal L supported by a non empty argument $\mathcal{A}$
- ***Derived belief***: it is a strict or a defeasible belief.

In Fig.1 the BR are the rules used to get the ***Derived beliefs***.

Example 2 *The **Derived beliefs** of the agent of Fig.1 will be* $\{b_2\}$ *, because there is an undefeated argument* $(b_2 \text{—}\prec s_3)$ *for literal* b_2*. Thus, this agent will believe in* $\{s_1, s_3, s_4, s_5, b_2\}$

The BDI-DeLP does not specify any mechanism to manipulate beliefs. However two processes can be identified: one to update the ***Perceived beliefs*** in every deliberative cycle; and other to add, modify or remove beliefs rules from $\mathcal{P}_B$, when the agent decides to do so.

In 3APL agent beliefs can be specified by any logic language [5]. However, as it was exposed in Section 2.2 3APL is an agent programming language, so a logic language for beliefs must be given. In their works they use propositional logic or Prolog logic. In this work we will study the approach of 3APL that uses Prolog to specify beliefs (see 3APL platform [9]). Thus, agent beliefs are specified by a Prolog program called `BELIEF BASE`. Therefore, a Belief can be a fact or a literal derived by Prolog rules. The agent can consult about its beliefs using a Prolog query.

In 3APL beliefs are not categorized. However the categorization exposed in BDI-DeLP above can be partially emulated. To specify ***Perceived beliefs***, Prolog facts can be used. This can be seen

in Fig.2 with the fact `perception([s1, s3, s4, s5])`, where the argument is the list of the perceived facts. ***Strict beliefs*** can be specified using Prolog rules. Using a Prolog program is not possible to specify ***Defeasible beliefs***, because Prolog does not have defeasible rules. Hence the ***Derived beliefs*** are only strict beliefs. Defeasible rules allow the programmer, for example to express easily exceptions of general rules (see [4] for examples).

It can be seen in Fig.1 and Fig. 2 that the representation of the agent of Ex.1 beliefs is different. One is made in Prolog and the other in DeLP. Form this difference, some issues arise. First, DeLP allows four types of answers for a query, whereas Prolog allows only two. Thus some kind of decision choice must be made in the Prolog approach, at the cost of losing flexibility in the agent reasoning process. It can be seen in the example of Fig.2 that the decision choice taken in this work is that UNDECIDED and UNKNOWN answers will be NO in Prolog. Nevertheless, this decision choice is not as restrictive as it seems, because it is considered reasonable to be skeptical about Beliefs. Second, a DeLP Program can sometimes be emulated by Prolog program. However, this emulated DeLP program will have more complex rules, will lose declarativity and will be less scalable. This comes because the Prolog rules will try to emulate the argumentative process done in DeLP. This can be seen in the form of the belief rules related to $b1$ in Fig.1 and Fig.2. Moreover, If the selection criterion of DeLP is changed or more rules related to $b1$ are added to BR in Fig.1, all the rules related to `b1` in the `BELIEF BASE` of Fig.2 should be revised and in the worst case all will be changed.

Example 3 *Consider the agents of Fig.1 and Fig.2. If the belief rule $\sim b_1 -\prec s_3$ is added to the $Beliefs\ Rules$ set of Fig.1, then the term `not(s3)` should be added to all rules related to `b1` in `BELIEF BASE` of Fig.2.*

Third, DeLP has mechanism to specify and use Strong Negation and Prolog does not. In brief, most of these issues come from the advantages DeLP over Prolog as knowledge representation languages. These advantages are exposed in [4].

3APL, as an programming language needs an explicit mechanism to add, remove or revise beliefs. For this, 3APL provides the set `CAPABILITIES`, which contains rules to add or remove beliefs. These rules, are called *basic actions* and can be used as a part of a plan. However, as it was said above, 3APL does not categorize beliefs. Therefore, all the beliefs are treated equally, which means that all the beliefs are updated using *basic action* rules. This means that the agent will update its perception beliefs only when *basic action* rules are executed, that is when it is executing a plan. Moreover, if the agent wants to update its perception beliefs every deliberation cycle, every plan it executes should include the *basic action* rules that update the perception (plans are explained in Section 3.1.3). This can be seen in Fig.2 where every rule in the `RULEBASE` set has, as part of its plan, the sequence `Sense(X),Add(X)`. Where `Sense(X)` is an external function to sense the environment and `Add(X)` is the basic action to add the acquired information `X` to the `BELIEF BASE`

3.1.2 Desires

In BDI-DeLP the set D is used to specify desires of the BDI theory. This set is composed of Literals, and can be contradictory. It represents all the desires that the agent want to achieve. A desire in this proposal represents an action that the agent wants to do. The agent only adopts a subset of desires, each time. This subset contains the current achievable desires and will be called $\mathbf{D}^c$. To do this a DeLP program $\mathcal{P}_F = (\Pi_F, \Delta_F)$ called *Filtering Rules* is used. This program $\mathcal{P}_F$ contains strict and defeasible rules that represent reasons for and against adopting desires (see Fig.1). To build the set $\mathbf{D}^c$, the DeLP program K_{ag}= (Π_{ag}, Δ_{ag}) is used. The program K_{ag} represents the agent knowledge base and it is created merging $\mathcal{P}_F$ and $\mathcal{P}_B$ (see [7] for details).

Example 4 *Consider the sets* Φ, BR *and* $\mathcal{P}_F$ *from Fig.1. The program* K_{ag}= (Π_{ag}, Δ_{ag}) *will be*

$$\Pi_{ag} = \left\{ \begin{array}{cc} s_1 & s_3 \\ s_4 & s_5 \end{array} \right\} \quad \Delta_{ag} = \left\{ \begin{array}{llll} b_1 -\!\prec s_1 & b_1 -\!\prec s_2 & \sim b_1 -\!\prec s_5 & \sim b_1 -\!\prec s_1, s_2 \\ b_2 -\!\prec s_3 & \sim b_2 -\!\prec s_2, s_4 & g_1 -\!\prec b_1 & \sim g_1 -\!\prec b_2 \\ g_2 -\!\prec s_3 & \sim g_2 -\!\prec \sim b_1 & & \end{array} \right\}$$

In Ex.4 the K_{ag} program is composed by all rules from BR and $\mathcal{P}_F$, and all the literals in Φ of Fig.1. Besides K_{ag} a *selection criterion T* is used. This criterion establishes the answers of K_{ag} that will be acceptable for the agent. Therefore, it determines the type of the agent. The elements in $\mathbf{D}^c$ will be only those elements of D that satisfy the selection criterion *T* in K_{ag}. To this purpose a filtering function is used. This function takes the selection criterion *T* and the desire set D, and builds $\mathbf{D}^c$. In [6] two possible selection criterion T are specified: one for *cautious agents*, that accepts only desires δ, if there is a warrant for δ from K_{ag}; and the other for *bold agents*, that accepts desires δ, if there is no warrant for $\sim\delta$ from K_{ag}. In the example of Fig.1 the agent type is not shown, but we assume that the agent is cautious, so he believes and desires only warranted things.

Example 5 *Consider a cautious agent A, with* K_{ag} *of the Ex. 4, and* D *of Fig.1. The set* $\mathbf{D}^c$ *set will be* $\{g_2\}$*, because there is an undefeated argument for* g_2 ($g_2 -\!\prec s_3$)*, and there is no warrant for* g_1.

In BDI-DeLP there is no mechanism to drop, modify or add new desires to the desire set D. However, the agent can have all the desires that will use in his lifespan in the set D, and use the K_{ag} to select which of them are currently are achievable. We assume that a agent can not adopt new desires. Thus the agent selects desires from the desire pool that are currently achievable under some policy.

In 3APL the `GOAL BASE` set is used to specify Desires of the BDI theory. This set is composed by terms, which are called goals. Goals represent the situation where the agent wants to be in. The goals in `GOAL BASE` are those goals that the agent want to achieve. Thus the `GOAL BASE` is similar to the desire set D of BDI-DeLP, which is exposed above. In 3APL there is no independent mechanism to determine which goals are currently achievable. Hence, this means that all goals in the `GOAL BASE` can be adopted as intentions.

In our opinion the agent development tool should provide some way to specify conditions over the desires. In the BDI-DeLP approach this is done via the filtering rules explained above. In 3APL, there are three alternatives for doing it, two indirectly via the *reasoning rules* (see 3.1.3), and the other at a meta level with the deliberative cycle program. One of those alternatives is using the goal rules (see 3.1.3), where the agent can manipulate the `GOAL BASE` to emulate the currently achievable desires set $\mathbf{D}^c$ of the other proposal. For us, this alternative leads the agent to have desires to change other desires. Other alternative is putting every condition for a goal as part of the precondition of the *reasoning rules* related to that goal. This is the alternative that was adopted in Fig.2. There, it can be seen that both reasoning rules related to the goal `g1` have, as part of their preconditions, the consults `b1, not(b2)`. This tries to emulate the conditions that are established by the filtering rules $\{\ g_1 -\!\prec b_1\ ,\ \sim g_1 -\!\prec b_2\ \}$ of the example from Fig.1. This design choice leads to some issues over the *reasoning rules* that will be detailed in the next section (because they are related to the intention model). Finally, the last alternative to expose conditions over goals in 3APL is using the tools to change the deliberative cycle program, which are proposed in [2]. These tools allow to redefine the deliberative cycle of the 3APL interpreter. In the redefinition the developer can put conditions related to the goals and the beliefs in the deliberative cycle. Thus the behavior of the filtering rules of the other approach can be emulated using these tools. These tools are at a meta level of the agent (The interpreter algorithm is changed) thus, every time a rule over the desires is to be added, the whole deliberative cycle algorithm must be revised. Therefore, compered to the other ways to specify the conditions over desires in 3APL this way is more unscalable. Thus, using one of these alternatives the filtering rules of BDI-DeLP can be emulated in 3APL. However, the same issues marked in Section 3.1.1 and exemplified in Ex.3 will arise.

In 3APL it is possible to define the agent type using the tools to redefine the delineative cycle program[2], exposing conditions over the execution of the mental components.

3.1.3 Intentions

In BDI-DeLP *intention rules* are used to specify how desires are adopted as intentions. An intention in this model is a current desire that the agent can commit with under some established preconditions and constrains. An *intention rule* will be applicable if its associated desire is in the $\mathbf{D}^c$ set and its preconditions and constrains are satisfied. Satisfied preconditions are warranted beliefs of K_{ag} or desires in $\mathbf{D}^c$. Satisfied constrains are not warranted beliefs of K_{ag} or desires that are not in $\mathbf{D}^c$. To select between various applicable intention rules, a selection policy is used. This policy can be specified by the developer. In the example of Fig.1 the intention rules are model by the set IR, and the policy used is *"Select the first applicable rule"*.

Example 6 *Consider the agent of Fig.1,* K_{ag} *of Ex.4 and* $\boldsymbol{D}^c$ *of Ex.5. The only applicable rule from* IR *of this agent will be* $g_2 \Leftarrow \{s_4\}\{\}$*, because* g_2 *is the only desire in the set* $\boldsymbol{D}^c$ *and* s_4 *is derivable from the program* K_{ag}*.*

The BDI-DeLP proposal does not specify a way to manipulate or write the actions associated with a selected intention, which is left to future works. Also in their approach there is no mechanism to add, modify or remove Intention Rules.

In 3APL intentions are specified by the set of rules called `RULEBASE`. The rules of this set are called *Reasoning Rules* and can be divided in three types: the *interaction rules*, the *goal rules* and the *plan rules*. The *plan rules* are used to revise or drop plans, and will not be treated in this work. The *interactions rules* and the *goal rules* are used to specify intentions.

Interaction rules of 3APL allow the agent to reach its goals via plans. These rules consist of a header goal, a guard, and a plan. The header goal is the goal which the agent will commit to. The guard represents the preconditions that must be satisfied (with respect to the `BELIEF BASE`) to apply the rule. The plan represents the set of actions that the agent will do to achieve the selected goal. Thus a rule of this set is a map under some established preconditions between a goal and a plan to solve it. The *goal rules* are very similar, in its form, to the interaction rules. The difference between these rules types is that the plans of the goal rules are only actions over the `GOAL BASE`. In the example of Fig.2 the set `RULEBASE` is only composed by *interaction rules* (No *goal* or *plan rules* were used).

In 3APL a interation/goal rule is applicable if its header goal is in the `GOAL BASE` and its guard is satisfied. To satisfy a guard all its elements must be derivable from the `BELIEF BASE` Prolog program. Once a rule is selected, its plan is executed.

Example 7 *Consider the 3APL agent of Fig.2. The only applicable interaction rule is* `g2 <- perceived(s4),perceived(s3),not(tail1)|{PlanGoal2,Sense(X),Add(X)}` *because g2 is in the* `GOAL BASE` *and the guard elements of the guard* `perceived(s4)`, `perceived(s3)` *and* `not(tail1)` *are derivable from the* `BELIEF BASE` *program. Note that, despite* `g1` *is in the* `GOAL BASE`*, all the interaction rules related to* `g1` *do not satisfy its guards.*

To select between two or more applicable rules, a policy can be used. In the 3APL Platform [9] a simple policy is used: Select the first specified rule (The same that we used in the example of BDI-DeLP). However, this policy can be redefined using the tools to redefine the deliberative cycle [2].

As it was mentioned above the guard of a 3APL interaction/goal rule will have all preconditions related to the applicability of an intention. However, it was exposed in 3.1.2 that the conditions over goals are added to the guard of the interaction/goal rule. Thus, the guard of the interaction/goal rule in 3APL will have preconditions to verify two semantically different things: if the rule is applicable; and if the associated goal is achievable. This issue add some undesired practical effects in the rules.

Example 8 *Consider the agent of Fig.2. It can be seen that* `b1, not(b2)`*, which represents the conditions for the* `g1`*, is repeated in every rule that has* `g1` *as a header goal.*

In general, all the conditions over a goal will be present in the guard of every interaction/goal rule related to that goal. This "repeated code problem" can become bigger if the conditions over the goals are more complex. These issues are not present in BDI-DeLP, because it provides different mechanisms to represent conditions over desires and preconditions over intention rules.

Contrary to BDI-DeLP, 3APL, as agent programming language has a well defined model to specify the actions (plans) associated with the intention, and rules (*plan rules*) to revise them. 3APL does not specify a mechanism to add, modify or remove reasoning rules.

3.2 Reasoning

In this section the deliberative cycle and the reasoning processes of both approaches are analyzed.

The BDI-DeLP approach does not explicitly define a deliberative cycle, it is left open for the developer. However, the deliberative cycle can not be arbitrary defined. There are several restrictions on how the cycle can be built. These restrictions are due to the way the sets for the reasoning process are used. Next, one possible deliberative cycle will be exposed:

1. Find which desires are currently achievable, using the K_{ag} DeLP program (beliefs and the filtering rules).
2. Find intention rules that match with the desires selected in step 1.
3. Find which rules, of those selected in step 2, satisfy the preconditions and constrains, using the K_{ag} DeLP program
4. If one or more rule satisfies step 3, then select one rule using the policy p.

The reasoning process of the BDI-DeLP proposal has two stages. One when the agent determines if a desire is in the set of the current desires, and the other when it checks if an intention rule is applicable. The applicable intention policy can also be thought as part of the reasoning process.

3APL, as an agent programming language has a concrete deliberative cycle specified [9],[2]. However, as it was stated in previous sections there are tools to reprogram the procedure that specifies the cycle [2]. The 3APL Deliberative cycle exposed in[2] is described next:

1. Find a reasoning rule matching with goals in the `GOAL BASE`.
2. Find which rules of step 1 satisfy its preconditions (Guard) with respect to the `BELIEF BASE` Prolog program.
3. If one or more rules satisfy steps 1 y 2, select a rule and execute its plan.

The reasoning process of 3APL has one stage, that is, when the agent checks if a rule is applicable or not. Considering what was stated in Section 3.1.3, this means that 3APL will test in the same reasoning step if a goal is achievable and if an interaction/goal rule is applicable. This leads to a practical issue. It was explained in Section 3.1.2 and 3.1.3, that all conditions for a goal will be in the guard of a rule that has that goal as a header goal. Thus the conditions for a goal will be tested in every rule that has that goal as a header goal. For example, it can be seen in Fig.2 that the conditions `b1, not(b2)` of the goal `g1` are not satisfied (`b1` is not derived from the `BELIEF BASE` program). These conditions will be tested with the guard of every rule related to `g1`. This means that these conditions will be tested two times in every deliberative cycle. This problem is not present in the BDI-DeLP approach because it does desire filtering and intention rule applicability tests in two separate steps. Thus, if 3APL would have two steps for those two stages the problem exposed above would not be present.

4 3APL WITH DELP AS A BELIEF LANGUAGE: 3APL-DELP

In the last section an analysis of each approach was shown and there, strong points of each approach were highlighted. In this section some of these strong points are used to present and define a combined approach called 3APL-DeLP.

In the presentation of 3APL [5], the authors did not fix the logic language used for knowledge representation. To develop the rest of their works [2], [10], [3] they used propositional logic or Prolog as language to represent beliefs. In particular the 3APL interpreter[9] uses Prolog. In this work we propose a to use DeLP as the knowledge representation language in 3ALP. This will be done using a DeLP program as the `BELIEF BASE` program of 3APL. This proposal will be called 3APL-DeLP.

A 3APL-DeLP agent will have the same mental components as a 3APL agent. Thus the mental state (configuration) of 3APL-DeLP is the same given in[3]. Next, we will define the components of 3APL that are affected by our proposal.

It was shown in Section 3, that DeLP is a more sophisticated mechanism for knowledge representation than Prolog. Thus, a the program used to represent beliefs of 3APL-DeLP must be defined as follows

Definition 1 : ***(*Belief Base*)*** *The `BELIEF BASE` of a 3APL-DeLP Agent will be a DeLP program $\mathcal{P}_{BF} = (\Pi_{BF}, \Delta_{BF})$, where Π_{BF} is a non contradictory set of facts and Δ_{BF} is a set of defeasible rules.*

All the Beliefs that the agent believes are derivable by the `BELIEF BASE` program, therefore:

Definition 2 : ***(*Belief*)*** *Let $\mathcal{P}_{BF}$ a be `BELIEF BASE`, a Literal $\mathcal{B}$ is a Belief from $\mathcal{P}_{BF}$ iff $\mathcal{B}$ is warranted from $\mathcal{P}_{BF}$.*

In 3APL basic action rules can be used to add or remove beliefs from the Belief Program. Therefore:

Definition 3 : ***(*Basic Action Rule*)*** *A Basic Action rule Ba is an ordered triplet Ba=(Pre, $Name$, Pos), where: Pre=$\{P_1, \ldots, P_m\}$ is a set of Literals representing a preconditions for Ba, $Name$ is a predicate, and Pos=$\{X_1, \ldots, X_n\}$ is a consistent set of extended Literals representing the effects of executing Ba.*

All the basic action rules are in the `CAPABILITIES` set of the 3APL-DeLP agent. The predicate $Name$ of Def.3, is used to call capability rule from the plans. This predicate can contain variables which will be treated as in-mode parameters.

Definition 4 : ***(*Applicable Basic Action Rule*)*** *Let Ba be a Basic Action Rule in `CAPABILITIES` and $\mathcal{P}_{BF}$ a be `BELIEF BASE`. Ba will be applicable if every P_i in Pre is a Belief.*

Definition 5 : ***(*Basic Action Rule Effect*)*** *Let Ba be an Applicable Basic Action Rule in CAPABILITIES and $\mathcal{P}_{BF}$ a be `BELIEF BASE`. The effect of executing Ba is is the revision of the `BELIEF BASE` DeLP program. This revision will consist of removing any literal in $\mathcal{P}_{BF}$ that is complementary of any literal in Pos or that is preceded by a "not" in Pos, and then adding all the literals in Pos to the resulting program.*

In 3APL an Interaction Rule of the `RULEBASE`, as it was explained in Section 3.1.3, involve a goal, a guard and a plans. The plan is the specifies the actions needed to solve the goal and the guard are the preconditions of the rule. Those preconditions are queries to the `BELIEF BASE`. Therefore:

Definition 6 : ***(*Interaction Rule*)*** *A Interaction rule R is an ordered triplet R=(HG, $Guard$, Pl), where: HG is a Literal representing the goal to achieve, $Guard$=$\{G_1, \ldots, G_n\}$ is a consistent set of extended Literals representing the preconditions for R, and Pl is a plan.*

Note that the definition of *goal rules* is analogous to Def.6, there, a goal action should be in the place of the plan.

Definition 7 : ***(*Applicable Interaction Rule*)*** *Let R be a Interaction rule in `RULEBASE` and $\mathcal{P}_{BF}$ a be `BELIEF BASE`. R will be applicable if HG is in `GOALBASE` and each G_i in $Guard$ is a Belief.*

Thus, using DeLP as the BELIEF BASE program allows to model the rules to represent knowledge in a more declarative way. DeLP characteristics allows to model incomplete and potentially contradictory information. The inference mechanism of DeLP allows the 3APL-DeLP agent to decide between several contradictory conclusions, and to adapt to changing environments, *i.e.*, it allows to add or remove information in a dynamic way, without the need of changing every rule of the agent, which adds scalability and flexibility. More examples of DeLP as a knowledge representation language are shown in [4]. Next, in Fig.3, a complete specification of the agent of Ex.1 is shown.

```
CAPABILITIES {
    {perception(Y)} Add(X) {not perception(Y), perception(X)}
} BELIEFBASE {
   perception([s1, s3, s4, s5]),
   perceived(X) -< perception(L),member(X,L).
   b1 -< perceived(s1).
   b1 -< perceived(s2).
   ~b1 -< perceived(s5).
   ~b1 -< perceived(s1),perceived(s2).
   b2 -< perceived(s3).
   ~b2 -< perceived(s2),perceived(s4).
} GOALBASE {
    g1, g2
} RULEBASE {
    g1 <- perceived(s1), perceived(s4), b1, not(b2) | {Plan1Goal1, Sense(X), Add(X)},
    g1 <- perceived(s4), b1, not(b2) | {Plan2Goal1, Sense(X), Add(X)},
    g2 <- perceived(s4), perceived(s3), not(~b1) | {PlanGoal2, Sense(X), Add(X)},}
```

Figure 3: 3APL-DeLP Agent example

As shown in Fig.3 the BELIEF BASE is a DeLP program is similar of the BR and Φ of Fig.1. The only difference between these two DeLP programs is that perceptions in Fig.3 were specified in the same way of the 3APL agent of Fig.2. It can be seen that new rules can be added without the need of changing all the BELIEF BASE program. Thus 3APL-DeLP will allow to specify BDI based agents with the benefits of the declarativity, planning support and communicative support of 3APL and all the benefits of DeLP as a knowledge representation language.

5 CONCLUSIONS AND FUTURE WORK

In this work we have shown 3APL-DeLP which is an approach to a programming Language for BDI Agents that uses defeasible argumentation for agent knowledge representation and reasoning. To reach this programming language two approaches based on BDI theories were studied. 3APL a programming language for cognitive agents and BDI-DeLP an BDI agent architecture that uses DeLP for knowledge representation and reasoning.

First, an introduction for each approach was given. There, goal, scope and advantages of each were characterized, and an example of an agent specified in each approach was shown. Then an analysis of the knowledge representation and reasoning features of each approach was developed. This analysis was divided in the Belief, Desires, Intention and reasoning of each approach. There, the advantages of DeLP as knowledge representation were characterized and shown with examples.

Finally, considering the advantages of DeLP exposed in the analysis of both approaches, the advantages of 3APL as agent programming language and the fact that in 3APL the knowledge representation language is not fixed, 3APL-DeLP was proposed. In 3APL-DeLP the agent knowledge is represented by a DeLP-program and the agent may reason with a defeasible argumentation formalism. Definitions and an example for this programming language were given. Thus, 3APL-DeLP allows the

specification of BDI based agents with the benefits of the declarativity, planning support and communicative support of 3APL and all the benefits of DeLP as a knowledge representation language.

In this work we only gave the definition of those components of 3APL that were affected by our proposal, however a full revision of other components is needed. Also, further research in how planning is affected by our proposal should be done. Finally, we think that filtering rules can be added to 3APL-DeLP, this has been left for future research.

REFERENCES

[1] M. E. Bratman, D. Israel, and M. Pollack. Plans and resource-bounded practical reasoning. In Robert Cummins and John L. Pollock, editors, *Philosophy and AI: Essays at the Interface*, pages 1–22, Cambridge, Massachusetts, 1991. The MIT Press.

[2] M. Dastani, F. de Boer, F. Dignum, and J. Meyer. Programming agent deliberation: An approach illustrated using the 3apl language. In *The Second Conference on Autonomous Agents and Multi-agent Systems (AAMAS'03), Melbourne, Australia*, 2003.

[3] M. Dastani, B. van Riemsdijk, F. Dignum, and J. Meyer. A programming language for cognitive agents: Goal-directed 3apl. In *First Workshop on Programming Multiagent Systems: Languages, frameworks, techniques, and tools (ProMAS03), Melbourne, Australia*, 2003.

[4] A. Garcia and G. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(1-2):95–138, 2004.

[5] K. Hindriks, F. de Boer, W. van der Hoek, and J.-J. Ch. Meyer. Agent programming in 3apl. *Autonomous Agents and Multi-Agent Systems*, 2(4):357–401, 1999.

[6] N. D. Rotstein, A. J. García, and G. R. Simari. Reasoning from desires to intentions: A dialectical framework. In *22nd. AAAI Conference on Artificial Intelligence*, July 2007.

[7] N. D. Rotstein, A. J. García, and Guillermo R. Simari. Defeasible reasoning about beliefs and desires. In *11th International Workshop on Non-Monotonic Reasoning (NMR 2006)*, pages 429–436. Lake District, UK, May 2006.

[8] Y. Shoham. Agent-oriented programming. In *Artificial Intelligence*, 1993.

[9] E.C. ten Hoeve. 3apl platform. Master's thesis, Univ. of Utrech, 2003.

[10] J. van der Ham. Multi-agent fipa compliant 3apl agents. Master's thesis, Univ. of Utrech, 2002.

[11] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

Máquinas de Vectores Soporte Adaptativas

Guillermo Grinblat y Alejandro Ceccatto

Centro Internacional Franco-Argentino de Ciencias de la Información
y de Sistemas (CIFASIS) - Universidad de Marsella/UNR/CONICET
Bvd. 27 de Febrero 210 Bis, 2000 Rosario, República Argentina

Resumen Se propone un método de clasificación adaptativo capaz de aprender un concepto y seguir su evolución temporal como consecuencia de cambios lentos en sistemas evolutivos. Para ello se realiza una modificación del clasificador SVM (máquina de vectores soporte), consistente en usar múltiples hiperplanos válidos en pequeñas localidades temporales (ventanas) para realizar la clasificación. A diferencia de otras propuestas de este tipo en la literatura, en este caso se realiza un aprendizaje de todos los hiperplanos en forma global, minimizando una cantidad que contiene al error que comete la familia de clasificadores locales más una medida asociada a la dimensión VC de los mismos. Para conceptos estacionarios, la misma idea aplicada a localidades en el espacio de características permite obtener resultados comparables a los que proporciona SVM con kernel gausiano.

1. INTRODUCCIÓN

Muchos problemas de clasificación asociados a sistemas del mundo real varían con el tiempo. Por ejemplo, un sistema puede variar por razones físicas como las estaciones del año, o puede ser necesaria su adaptación por cambio en las expectativas o intereses de los usuarios del mismo. En la mayoría de los casos, la causa y características de este cambio no se presentan en los datos a analizar de manera obvia, por lo que el clasificador asociado tiene que ser capaz no sólo de aprender la relación entrada-salida correcta en cada instante de tiempo sino además advertir el cambio en el concepto y adaptarse a él.

Habitualmente este problema se trata usando una ventana temporal, suponiendo que el cambio en el concepto a aprender es despreciable dentro de ella [1]. Si la ventana es de ancho muy grande, dicha suposición en general no es válida y el tiempo de adaptación del algoritmo resulta excesivo. Por el contrario, cuando el ancho de la ventana es chico el algoritmo se adapta rápidamente, pero es más sensible al ruido y se torna impreciso ya que debe aprender la relación entrada-salida a partir de unos pocos ejemplos. Dentro de esta aproximación al problema, existen algoritmos que usan un ancho de ventana adaptable [1], donde se dividen los datos en grupos ("batchs") y se usa como ventana la cantidad de grupos óptima. Aún así, es igualmente necesario suponer que el concepto no cambia dentro de cada grupo de datos. Por otro lado, un enfoque puramente estadístico del aprendizaje de conceptos evolutivos puede hallarse en [2].

En este trabajo presentamos una nueva forma de encarar este problema, en la cual los clasificadores sucesivos varían siguiendo la mutación del concepto pero su ajuste se realiza de manera global. Es particular, la ventana temporal de validez de un clasificador puede ser tan chica como se desee (inclusive un sólo punto), pero los clasificadores se entrenan minimizando una medida global de error en lugar de ajustarse localmente. Esta filosofía se aplica onsiderando una adaptación de uno de los métodos de clasificación más potentes y estudiados en la actualidad, las llamadas "maquinas de vectores soporte" o SVM por sus siglas en inglés [3].

En el caso de clasificación estacionaria, la filosofía arriba descripta puede aplicarse reemplazando las ventanas temporales por vecindades de un punto en el espacio de características del problema. De esta forma, el clasificador SVM puede describir fronteras de decisión complejas (no simples hiperplanos) sin necesidad de apelar al truco del kernel ("kernel trick")[3] para lograr un clasificador no lineal. Mostraremos, a través de un ejemplo simple, que con esta implementación de SVM adaptativo es posible alcanzar resultados equivalentes a SVM con kernel gausiano.

2. JUSTIFICACIÓN DEL MÉTODO

Supongamos que tenemos un conjunto de datos $[(x_1, y_1) \ldots, (x_n, y_n)]$, donde (x_i, y_i) fue obtenido en el instante $t = i$ y $y_i = \pm 1$. Definimos:

Clase F: Clase de clasificadores tal que $f \in F$ implica que f implementa una frontera de decisión constituida por hiperplanos que cambian con t. Así, un punto en un instante dado es clasificado de acuerdo al lado del hiperplano correspondiente a ese instante en que se encuentra. Note que la dimensión de Vapnik-Chervonenkis (VC) [4] de F es ∞, ya que si el hiperplano cambia lo suficiente desde el instante $i-1$ al instante i, podrá clasificar bien el punto x_i sin importar dónde se encuentre.

Clase F *reducida*: Sea $f \in F_v$ un clasificador perteneciente a F, tal que el cambio del hiperplano de un instante al siguiente está acotado por v. Para ser más precisos, si en el instante i el hiperplano es $f_i = w_i \cdot x_i + b_i$, entonces

$$f \in F_v \Leftrightarrow (w_{i-1} - w_i)^2 + (b_{i-1} - b_i)^2 \leq v^2 \quad \forall i. \tag{1}$$

Sea $F_{v'}$ el conjunto de clasificadores que no cambian más que v', con $v \leq v'$. Como $F_v \subseteq F_{v'}$, la dimensión VC de $F_v \leq$ la dimensión VC de $F_{v'}$. Por otro lado, la dimensión VC de $F_{v=0}$ en $\Re^n$ es $n+1$, ya que es el conjunto de hiperplanos que no varían. Es decir, la dimensión VC de F_v crece con v.

De acuerdo a esto, si queremos controlar la complejidad de las funciones f podemos limitarnos a elegir funciones de F_v para cierto v. O, por la teoría de regularización, en vez de buscar la función de F_v que minimice cierto error $Err(f, x, y)$, podemos buscar la función de F que minimice

$$Err(f, x, y) + C \text{ comp}(f)$$

donde C es una constante que define la importancia relativa de los errores de clasificación con respecto a la complejidad de la función $\text{comp}(f) = \text{máx}_i[(w_{i-1} - w_i)^2 + (b_{i-1} - b_i)^2]$.

Alternativamente, podemos hacer el mismo razonamiento definiendo F_v como la clase de hiperplanos que en promedio se mueven menos que v, con lo que llegaríamos a minimizar

$$Err(f, x, y) + C\frac{1}{n}\sum_i (w_{i-1} - w_i)^2 + (b_{i-1} - b_i)^2$$

Por un razonamiento similar al realizado anteriormente, podemos llegar a la conclusión de que la dimensión VC baja cuando el margen promedio del hiperplano móvil crece. Definimos entonces la complejidad de f como

$$\text{comp}(f) = \frac{1}{n}\sum_i w_i^2 + \frac{C_2}{n}\sum_i (w_{i-1} - w_i)^2 + (b_{i-1} - b_i)^2$$

donde C_2 indica a qué causa de aumento de la dimensión VC le damos más importancia.

Definiendo ahora

$$Err(f, x, y) = \sum_i \text{máx}[0, 1 - y_i(w_i x_i + b_i)],$$

llegamos al problema

$$\text{mín}\, C_3 \sum_i \text{máx}(0, 1 - y_i(w_i x_i + b_i)) + \frac{1}{n}\sum_i w_i^2 + \frac{C_2}{n}\sum_i (w_{i-1} - w_i)^2 + (b_{i-1} - b_i)^2,$$

que es equivalente a

$$\text{mín}\, C_3 \sum_i \xi_i + \frac{1}{n}\sum_i w_i^2 + \frac{C_2}{n}\sum_i (w_{i-1} - w_i)^2 + (b_{i-1} - b_i)^2,$$

sujeto a

$$\begin{aligned} \xi_i &\geq 0 \\ y_i(w_i x_i + b_i) - 1 + \xi_i &\geq 0\ . \end{aligned}$$

3. SVM ADAPTATIVO

Tal como fuera explicitado más arriba, tenemos n puntos, $x_1 \ldots x_n$, divididos en dos clases, con $y_i = \pm 1$ la clase del punto x_i. Definimos ahora V_i como el conjunto de puntos vecinos a x_i y denotamos N_i al número de puntos en V_i. En lo sucesivo llamaremos M_{i*} a la fila i de la matriz M y M_{*j} a la columna j de dicha matriz. Con $P * Q$ indicaremos el producto miembro a miembro de las matrices P y Q; es decir, $(P * Q)_{ij} = P_{ij} Q_{ij}$.

Partimos del problema

$$\min_{w_i,b_i} \frac{1}{2n}\sum_{i=1}^{n}\left(||w_i||^2 + \frac{C_2}{2}\sum_{j\in V_i}||w_i - w_j||^2 + (b_i - b_j)^2\right) + C_3\sum_i \xi_i,$$

con $||w|| = w \cdot w$, sujeto a

$$\begin{aligned} \xi_i &\geq 0 \\ y_i(w_i x_i + b_i) - 1 + \xi_i &\geq 0. \end{aligned}$$

El correspondiente lagrangeano es:

$$\begin{aligned} L = \frac{1}{2n}\sum_{i=1}^{n}\left(||w_i||^2 + \frac{C_2}{2}\sum_{j\in V_i}||w_i - w_j||^2 + (b_i - b_j)^2\right) + C_3\sum_i \xi_i \\ - \sum_i \alpha_i(y_i(w_i x_i + b_i) - 1 + \xi_i) - \sum_i \beta_i \xi_i, \end{aligned} \tag{2}$$

donde $\alpha_i \geq 0$ y $\beta_i \geq 0$.

Tenemos que maximizar L con respecto a los α_i y β_i y minimizarlo con respecto a los w_i, b_i y ξ_i. En este punto de mínimo, las derivadas con respecto a las variables primales tienen que ser nulas:

$$\frac{\partial L}{\partial \xi_i} = 0 = C_3 - \alpha_i - \beta_i,$$

lo cual implica que

$$0 \leq \alpha_i \leq C_3.$$

Por otro lado, teniendo en cuenta que cada ξ_i está en L multiplicado por $C_3 - \alpha_i - \beta_i$, (2) queda

$$L = \frac{1}{2n}\sum_{i=1}^{n}\left(||w_i||^2 + \frac{C_2}{2}\sum_{j\in V_i}||w_i - w_j||^2 + (b_i - b_j)^2\right) - \sum_i \alpha_i(y_i(w_i x_i + b_i) - 1). \tag{3}$$

En el caso de los w_i se tiene

$$\frac{\partial L}{\partial w_i} = 0 = \frac{1}{n}\left(w_i + C_2\sum_{j\in V_i}(w_i - w_j)\right) - \alpha_i y_i x_i,$$

donde se ha considerado que si x_j es vecino de x_i, x_i es vecino de x_j. Esta ecuación resulta

$$\frac{1}{n}\left((1 + C_2 N_i)w_i - C_2\sum_{j\in V_i} w_j\right) = \alpha_i y_i x_i. \tag{4}$$

Si definimos la matriz M y el vector z como

$$M_{ij} = \begin{cases} (1 + C_2 N_i)/n \text{ si } i = j \\ -C_2/n \qquad \text{si } j \in V_i \\ 0 \qquad\qquad \text{caso contrario.} \end{cases}$$

$$z_i = \alpha_i y_i x_i$$

podemos poner (4) en la forma $Mw = z$, o bien $w_i = (M^{-1})_{i*} z$. Reemplazando en (3) obtenemos

$$L = \frac{1}{2n}\alpha^T((M^{-1})^2 * K)\alpha + \frac{C_2}{4n}\alpha^T((M^{-1}QM^{-1}) * K)\alpha + \frac{C_2}{4n}\sum_i \sum_{j \in V_i} (b_i - b_j)^2$$
$$-\alpha^T((M^{-1}) * K)\alpha + \sum_i \alpha_i - \sum_i \alpha_i y_i b_i. \tag{5}$$

donde hemos definido dos matrices $N \times N$, $K_{ij} = y_i y_j x_i x_j$ y Q dada por

$$Q_{ij} = \begin{cases} N_i \text{ si } i = j \\ -1 \text{ si } i \text{ es vecino de } j \\ 0 \quad \text{caso contrario} \end{cases}$$

En el caso de los b_i,

$$\frac{\partial L}{\partial b_i} = 0 = \frac{C_2}{n}\sum_{j \in V_i}(b_i - b_j) - \alpha_i y_i,$$

de lo que obtenemos

$$\frac{C_2 N_i}{n} b_i - \frac{C_2}{n}\sum_{j \in V_i} b_j = \alpha_i y_i. \tag{6}$$

Definiendo h como:

$$h = \begin{pmatrix} \alpha_1 y_1 \\ \vdots \\ \alpha_n y_n \end{pmatrix}$$

podemos escribir (6) en la forma

$$\frac{C_2}{n} Qb = h. \tag{7}$$

Dado que Q es singular,

$$Q\overline{1} = \begin{pmatrix} N_1 - \sum_{j \in V_i} 1 \\ \vdots \\ N_n - \sum_{j \in V_n} 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

se puede afirmar que

$$0 = \frac{C_2}{n}\overline{0}b = \frac{C_2}{n}\overline{1}Qb = \overline{1}h = \sum_i \alpha_i y_i.$$

3.1. Eliminación de los b_i

Es posible eliminar b de (5). La parte que depende de b es

$$\frac{C_2}{4n}\sum_i\sum_{j\in V_i}(b_i-b_j)^2-\sum_i\alpha_i y_i b_i=\frac{C_2}{4n}\sum_i\sum_{j\in V_i}(b_i-b_j)^2-h^Tb.$$

Por otro lado,

$$b^TQb=\sum_i\sum_{j\in V_i}(b_i-b_j)b_i.$$

con lo que

$$\frac{C_2}{4n}\sum_i\sum_{j\in V_i}(b_i-b_j)^2-h^Tb=\frac{C_2}{2n}b^TQb-h^Tb.$$

De (7) sabemos que

$$b^TQ=\frac{nh^T}{C_2},$$

con lo cual queda

$$\frac{C_2}{2n}\frac{nh^T}{C_2}b-h^Tb=-\frac{h^Tb}{2}.$$

Diagonalizando Q resulta $Q=PDP^T$, con $P^{-1}=P^T$ ya que Q es simétrica, con lo cual de (7) podemos obtener

$$DP^Tb=P^T\frac{nh}{C_2}$$

Si definimos $b'=P^Tb$ y $h'=P^Th$, esto es

$$Db'=\frac{nh'}{C_2}.$$

Como D es diagonal, es fácil encontrar los b'_i para los cuales $D_{ii}=\lambda_i\neq 0$:

$$b'_i=\frac{nh'}{C_2\lambda_i},\text{ si }\lambda_i\neq 0.$$

Si el autovalor $\lambda_i=0$, de (7) tenemos que

$$\begin{aligned}P_{i*}^TQb&=P_{i*}^T\frac{nh}{C_2}\\ \overline{0}b&=P_{i*}^T\frac{nh}{C_2}\\ 0&=\frac{nh'_i}{C_2}.\\ 0&=h'_i\end{aligned}$$

Es decir, cuando no podemos determinar b'_i por la ecuación anterior, $h'_i = 0$.

Volvamos a lo que queremos obtener:

$$\begin{aligned} -\frac{h^T b}{2} &= -\frac{h^t P P^T b}{2} \\ &= -\frac{1}{2} h'^T b' \\ &= -\frac{1}{2} \sum_i h'_i b'_i \\ &= -\frac{1}{2} \sum_i \frac{n}{C_2} \Lambda_i h'^2_i \end{aligned}$$

donde $\Lambda_i = \frac{1}{\lambda_i}$ si $\lambda_i \neq 0$ y $\Lambda_i = 0$ si $\lambda_i = 0$. Si se define D' como la matriz diagonal $D'_{ii} = \Lambda_i$ e Y dada por $Y_{ij} = y_i y_j$, esto último queda

$$\begin{aligned} -\frac{h^T b}{2} &= -\frac{n}{2C_2} h^T P D' P^T h \\ &= -\frac{n}{2C_2} \alpha^T ((P D' P^T) * Y) \alpha. \end{aligned}$$

Reemplazando en (5) obtenemos

$$\begin{aligned} L = \frac{1}{2n} \alpha^T ((M^{-1})^2 * K) \alpha + \\ \frac{C_2}{4n} \alpha^T ((M^{-1} Q M^{-1}) * K) r r \alpha - \\ \alpha^T ((M^{-1}) * K) \alpha + \sum_i \alpha_i - \\ -\frac{n}{2C_2} \alpha^T ((P D' P^T) * Y) \alpha \end{aligned}$$

Es decir,

$$L = \alpha^T \left[\left(\frac{1}{2n} M^{-2} + \frac{C_2}{4n} M^{-1} Q M^{-1} - M^{-1} \right) * K - \frac{n}{2C_2} (P D' P^T) * Y \right] \alpha + \sum_i \alpha_i,$$

lo que implica que es de la forma

$$L = \alpha^T R \alpha + \sum_i \alpha_i$$

con la matriz R definida de manera adecuada.

El problema dual queda entonces

$$\max_\alpha \alpha^T R \alpha + \sum_i \alpha_i,$$

sujeto a

$$0 \leq \alpha_i \leq C_3$$
$$\sum \alpha_i y_i = 0,$$

que es el problema resuelto en SVM (aunque con otra R). En consecuencia, para resolverlo se puede usar cualquiera de las técnicas empleadas para SVM convencional, como por ejemplo SMO[3].

4. EXPERIMENTOS

4.1. Bases de Datos Artificiales

Se probó el método en las siguientes tres bases de datos artificiales:

Dataset 1: Se generó un dataset de puntos $x_i \in \Re^2$, divididos en dos clases, la clase 1 formada por 250 puntos centrados en $(0, 1)$, con un desvío estandar de 0,1 en cada dimensión y la clase -1 formada por 250 puntos centrados en $(0, -1)$, con el mismo desvío.

Para simular un cambio a través del tiempo, a cada punto x_i, representado en coordenadas polares, se le sumó un ángulo de $\frac{i\pi}{500}$ radianes. Al algoritmo se le presentaron los puntos en coordenadas rectangulares. El resultado se grafica en la figura 1.

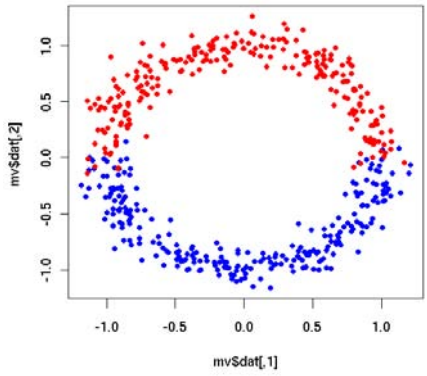


Figura 1. Dataset 1

Dataset 2: A los mismos puntos que en el dataset anterior se les sumó un ángulo $\frac{3i\pi}{1000}$ en vez de $\frac{i\pi}{500}$. En este caso el solapamiento de las clases es mayor (ver figura 2).

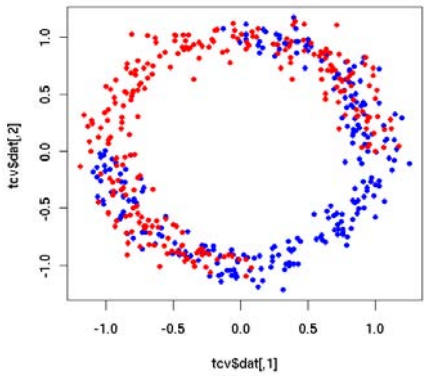


Figura 2. Dataset 2

Dataset 3: Se tomaron 500 puntos $x \in \Re^2$ al azar con $x_1 \in [-0{,}2; 1{,}2]$ y $x_2 \in [-5; 5]$, con distribución uniforme. Al punto x_i se le asignó la clase 1 si

$$x_{2,i} \geq \frac{1}{1 + e^{-x_{1,i}}}$$

y la clase -1 en caso contrario (ver Figura 3).

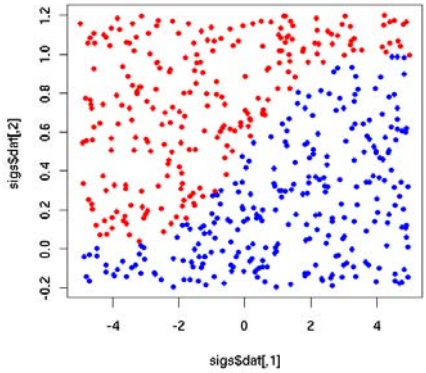


Figura 3. Dataset 3

La intención de las pruebas con los dos primeros datasets fue ver cómo evoluciona el clasificador a través del tiempo al variar el parámetro C_2 y compararlo con la solución encontrada por SVM estándar con kernel lineal (que no tiene en cuenta el tiempo en que cada una de las muestras fue colectada).

Para que el método busque una evolución temporal, se consideró como conjunto de puntos vecino a x_i a $\{x_{i-1}, x_{i+1}\}$ (salvo en el caso de los extremos). Es decir, se usó una matriz Q dada por

$$\begin{aligned} Q_{11} = Q_{nn} &= 1 \\ Q_{ii} &= 2 \quad \text{para } i \neq 1 \text{ e } i \neq n \\ Q_{i,i+1} = Q_{i,i-1} &= -1 \\ Q_{ij} &= 0 \quad \text{en otros casos.} \end{aligned}$$

El parámetro C_3 se dejó fijo en 1 mientras se variaba C_2. Acorde a esto, se usó $C = 1$ para el parámetro de SVM.

Los resultados obtenidos muestran que el clasificador cambia de la misma forma en que se generaron los puntos para los valores más bajos de C_2 y tiende a la solución encontrada por SVM estándar a medida que crece C_2. El error de entrenamiento es 0, salvo para valores altos de C_2, que obligan a que el clasificador no varíe mucho y en consecuencia no puede seguir la evolución de los datos (cabe aclarar que los datasets no tienen puntos mal clasificados). En la figura 4 se ve la evolución del clasificador para $C_2 \in \{10, 10^3, 10^6, 10^8\}$. Las líneas unen el extremo de cada vector w_i con el de los vectores w_{i+1} y w_{i-1}; es decir, constituyen la gráfica de la curva $w(t)$. Los errores de entrenamiento para estos cuatro casos fueron 0, 0, 0 y 2,6 %, respectivamente.

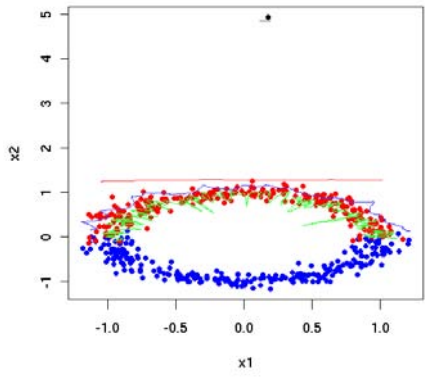


Figura 4. Dataset 1. Las líneas muestran la evolución $w(t)$ para $C_2 = 10$ (verde), 10^3 (azul), 10^6 (roja) y 10^8 (negra); el punto negro corresponde al w obtenido con SVM no adaptivo.

Para el dataset 2 se obtuvieron resultados similares. En la figura 5 se ven las curvas correspondientes a $C_2 \in \{10^3, 10^6, 10^7, 10^8\}$. Los errores de entrenamiento fueron de 0, 0, 0,2 % y 28,8 % para cada uno de estos valores. Cabe destacar que el alto valor de error correspondiente a $C_2 = 10^8$ constituye aproximadamente el resultado esperado para SVM estándar (no adaptativo), debido a la alta superposición de las clases.

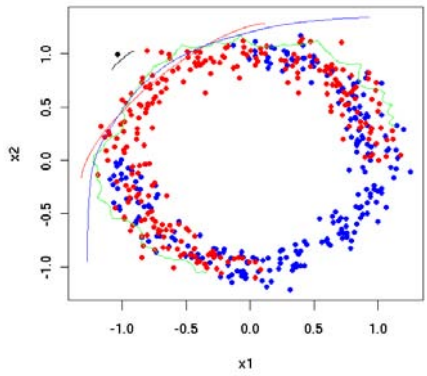


Figura 5. Dataset 2. Las líneas muestran la evolución $w(t)$ para $C_2 = 10$ (verde), 10^3 (azul), 10^6 (roja) y 10^8 (negra); el punto negro corresponde al w obtenido con SVM no adaptativo.

Con el tercer dataset se ejemplifica cómo se comporta el método en problemas cuya frontera de decisión no sea un simple hiperplano (tratando esta situación como un cambio adaptativo en el espacio de características del problema en lugar del tiempo). Si bien la frontera de decisión es una función compleja de la entrada, se pueden separar las clases con un hiperplano que varíe lentamente a medida que cambia una de las coordenadas. Note que en este caso la aplicación del método se realiza sobre un sistema estacionario, por lo que este ejemplo en realidad explora la capacidad del algoritmo aquí propuesto de constituirse en una alternativa al uso de SVM con kernels no-lineales.

Para que el método busque soluciones que varíen localmente se consideró como vecinos a los puntos x_i y x_j sii x_i es uno de los 5 vecinos más próximos a x_j o x_j es uno de los 5 vecinos más próximos a x_i. Lo mismo que antes, el parámetro C_3 se dejó fijo en 1 y se usó también 1 para el parámetro C de SVM. Como conjunto de test consideramos una grilla de puntos dentro del mismo rectángulo que los datos de entrenamiento. A cada punto de test se lo clasificó con el hiperplano asociado a su vecino más próximo dentro del dataset original usado para aprendizaje del clasificador.

En las figuras 6-9 se muestran los resultados obtenidos sobre el conjunto de test para valores de C_2 de 10^3, 10^4, 5×10^4 y 5×10^5 respectivamente. Como era esperable, para C_2 grande la frontera de decisión obtenida es muy similar a la que produciría el SVM estacionario con kernel lineal.

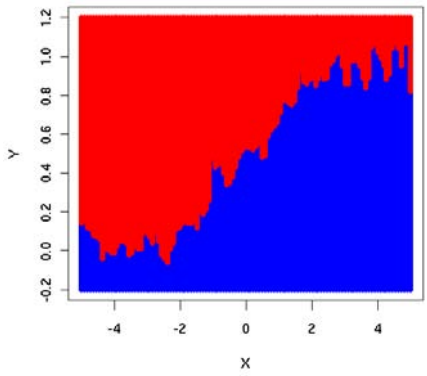


Figura 6. Dataset 3. Resultados para $C_2 = 10^3$.

También se compararon los resultados con los obtenidos con SVM con kernel gausiano. El reportado en el cuadro 1 es el mejor que se pudo conseguir variando el parámetro γ de la gausiana y dejando el parámetro C fijo en 1.

Cuadro 1. Errores de test para el dataset 3.

Prueba	Error (%)
$C_2 = 10^3$	3,61
$C_2 = 10^4$	2,12
$C_2 = 5 \times 10^4$	2,81
$C_2 = 5 \times 10^5$	5,17
SVM lineal	5,73
SVM gausiano ($\gamma = 2{,}94$)	2,37

Como puede verse en la tabla, nuestro método supera al resultado óptimo de SVM con kernel gausiano, aunque los resultados tienen que considerarse preliminares dado que para ninguno de los dos métodos se realizó un ajuste exhaustivo de los parámetros.

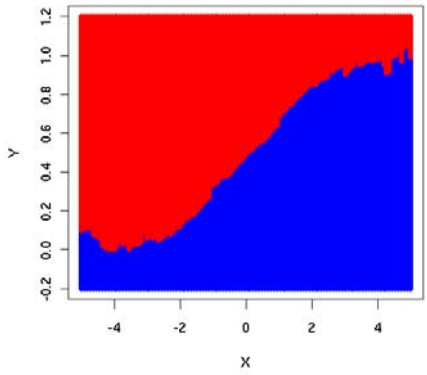


Figura 7. Dataset 3. Resultados para $C_2 = 10^4$.

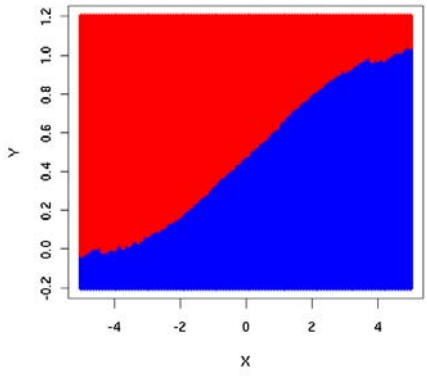


Figura 8. Dataset 3. Resultados para $C_2 = 5 \times 10^4$.

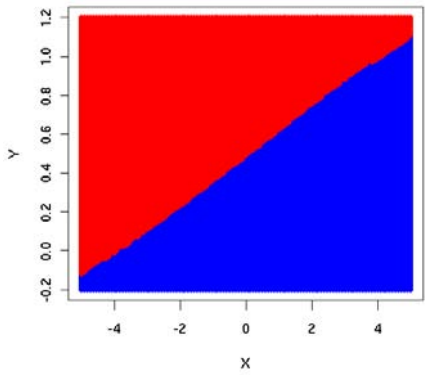


Figura 9. Dataset 3. Resultados para $C_2 = 5 \times 10^4$.

4.2. Bases de Datos Reales

De los dos primeros ejemplos sintéticos considerados en la sección anterior surge claramente la capacidad del algoritmo propuesto para resolver problemas de clasificación adaptativa. Ello no es sorprendente en tanto el método fue específicamente diseñado teniendo en mente este tipo de aplicaciones. Es destacable, en cambio, que el mismo permita resolver problemas de clasificación estándar con precisión comparable a la de SVM no lineal (Kernel SVM), tal como surge del análisis del tercer dataset considerado previamente. Para explorar más en detalle esta capacidad, se probó el método propuesto en un problema real. Para ello se usó el dataset **breast cancer** del *IDA repository* [5]. Al igual que en [6], se consideraron 100 divisiones de los datos para entrenamiento y test (200 para el entrenamiento y 77 para test). Para seleccionar los parámetros del método se hizo en cada una de las 100 pruebas una validación cruzada con 10 "folds". Se probó variar el parámetro C_2 en $\{10, 10^2, 10^3, 5 \times 10^3, 10^4, 5 \times 10^4\}$ y el número de vecinos usados en $\{3, 4, 5, 7, 10, 15, 20, 25\}$.

En el cuadro 2 se muestra el mejor resultado obtenido y el reportado en [6]:[1]

Cuadro 2. Errores de test para el dataset "breast cancer".

Prueba	Error (%)
Nuestro Método	$27{,}0 \pm 5{,}7$
SVM	$26{,}0 \pm 4{,}7$

[1] En [6] el σ reportado para SVM es de 0,47, mientras que en el resumen en [5] figura 4,7.

La pequeña diferencia en los promedios de error, que favorece al mejor resultado obtenido con SVM convencional, no es significativa dada la gran dispersión observada en dichos promedios. Es decir, nuevamente aquí, sobre un problema real, se obtiene que SVM adaptativo es comparable a SVM con kernel gausiano. Esta capacidad del método propuesto constituye un importante valor agregado del mismo, ya que permite generar un clasificador no lineal en el espacio de características original, independizándonos de la elección de un mapa a un espacio de mayor dimensión (a través del kernel utilizado).

5. CONCLUSIONES

En el presente trabajo hemos propuesto un nuevo método de generación de clasificadores adaptativos, capaces de aprender conceptos que mutan con el tiempo. La idea se ejemplificó desarrollando en detalle una versión local del clasificador SVM, que permite recuperar la evolución temporal de un problema de clasificación simple. La idea básica consiste en usar múltiples hiperplanos válidos en pequeñas localidades temporales (ventanas) para realizar la clasificación pero, a diferencia de otras propuestas de este tipo, realizando un aprendizaje de todos los hiperplanos en forma global. Para ello se minimiza una cantidad que contiene al error que comete la familia de clasificadores locales más una medida asociada a la dimensión de VC de los mismos. Por otro lado, la misma idea aplicada a localidades en el espacio de características del problema de clasificación permite obtener resultados comparables a los que proporciona SVM con kernel gausiano.

Los resultados presentados son preliminares y requieren aún una experimentación más exhaustiva para establecer ventajas y desventajas del método propuesto. No obstante, al menos para sistemas no estacionarios, los mismos son lo suficientemente prometedores como para ser optimista en cuanto a la aplicación de esta técnica a problemas reales en sistemas que mutan lentamente.

Referencias

1. R. Klinkenberg, T. Joachims, *Detecting Concept Drift with Support Vector Machines*, Proceedings of ICML-00, 17th International Conference on Machine Learning (2000)
2. Renato Vicente, Osame Kinouchi, and Nestor Caticha, *Statistical Mechanics of Online Learning of Drifting Concepts: A Variational Approach*, Machine Learning **32**, 179-201 (1998)
3. N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge (2000)
4. V. Vapnik, *Statistical Learning Theory*, Wiley (1998)
5. *IDA Benchmark repository used in several boosting, KFD and SVM papers* http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm
6. K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf, *An Introduction to Kernel-Based Learning Algorithms*, IEEE Transactions on Neural Networks 12, No. 2 (2001)

Random Forest-like strategies for Neural Network Ensembles Construction

Rafael Namías and Pablo M. Granitto
IFIR – CIFASIS, CONICET/UNR
Bv. 27 de Febrero 210 Bis, 2000 Rosario, Argentina
granitto@ifir.edu.ar

Abstract

Ensemble methods show improved generalization capabilities that outperform those of single learners. It is generally accepted that, for aggregation to be effective, the individual learners must be as accurate and diverse as possible. An important problem in ensemble learning is then how to find a good balance between these two conflicting conditions. For tree-based methods a successful strategy was introduced by Breiman with the Random-Forest algorithm. In this work we introduce new methods for neural network ensemble construction that follow Random-Forest-like strategies to construct ensembles. Using several real and artificial regression problems, we compare our new methods with the more typical Bagging algorithm and with three state-of-the-art regression methods. We find that our algorithms produce very good results on several datasets. Some evidence suggest that our new methods work better on problems with several redundant or noisy inputs.

Keywords: Machine Learning; Ensemble Methods; Neural Networks; Random Forest

1 Introduction

Over the last decade ensemble methods have been on the focus of machine learning research[10, 17]. The base of these procedures is the intuitive idea that by combining the outputs of several individual predictors one might improve on the performance of a single generic one. The so-called bias/variance dilemma [6] provides formal support to the success of these strategies. According to these ideas, good ensemble members must be both accurate and diverse, which poses the problem of generating a set of predictors with reasonably good individual performances and independently distributed predictions for the test points. As these are two opposite conditions, good ensemble methods achieve a compromise between them. Typical examples are bagging [2] and boosting [5].

Diverse individual predictors can be obtained in several ways. Bagging and boosting strategies are based on learning from different adequately-chosen subsets of the data set. Other methods try to generate diversity by adding small perturbations to the data at hand. For example, Rodriguez et al. [12] use partial PCA decompositions at each node of a CART Tree. Breiman [4] randomizes the class labels of a small sub-sample of the data when growing each tree and Martinez et al. [11] perform the same procedure for neural networks ensembles. Another successful strategy is to modify slightly the internal learning structure of a given algorithm in order to gain in diversity. For example, Geurts et al. [7] grow decision trees using randomly

selected variables and split points. Clearly, the most successful method of this class is the Random Forest (RF) algorithm [3], introduced by Leo Breiman. In RF, like in bagging, several CART trees are grown on bootstrap samples of the original dataset. But, when growing each tree, only a small random subset of features is considered at each node. Doing this, at each step the algorithm minimizes a cost function only in a randomly selected subspace of the full hypothesis space.

Several ensemble techniques have been recently applied to artificial neural networks (ANN) [8, 14, 16]. As the diversity of ANN comes naturally from the training process randomness and from the intrinsic non-identifiability of the model, it is difficult to improve over simple strategies like using several networks trained on the same data or plain bagging. For classification problems boosting of ANN outperforms other ensemble methods in several cases [16, 13] but for regression problems several methods show similar results [8]. Aiming at increasing the diversity in ANN ensembles, in this work we introduce new methods for neural network ensemble construction that follow Random-Forest-like strategies, i.e., that minimize the ANN cost function on randomly selected subspaces. We test the proposed algorithms on regression problems, using real and artificial benchmark datasets and time series.

This paper is organized as follows. In the next section we recall the bias/variance dilemma. Next, we introduce our new ensemble construction strategies. In Section 4 we describe the experimental settings used and in Section 5 we show and discuss the empirical results of our new methods. Finally, in Section 6 we draw some conclusions and discuss future lines of research.

2 Bias and Variance

In this section we will briefly recall the bias/variance decomposition of the generalization error [6] (which is the theoretical base of ensemble methods) following [9]. Let us consider a set of N noisy data pairs $D = \{(t_i, \mathbf{x}_i),\ i = 1, N\}$, where the vectors $\mathbf{x}_i$ of predictor variables are obtained from some distribution $P(\mathbf{x})$ and the regression targets t_i are generated according to

$$t_i = f(\mathbf{x}_i) + \varepsilon_i. \tag{1}$$

Here f is the true regression and ε is random noise with zero mean. If we estimate f using D obtaining a model f_D, the (quadratic) generalization error on a test point $(t, \mathbf{x})$ averaged over all possible realizations of the data set D (with respect to P and noise ε) can be decomposed as:

$$\mathrm{E}[(t - f_D(\mathbf{x}))^2|D] = \mathrm{E}[\varepsilon^2|\varepsilon] + (\mathrm{E}[f_D(\mathbf{x})|D] - f(\mathbf{x}))^2 + \mathrm{E}[(f_D(\mathbf{x}) - \mathrm{E}[f_D(\mathbf{x})|D])^2|D] \tag{2}$$

The first term on the right-hand side is simply the noise variance σ_ε^2; the second and third terms are, respectively, the squared bias and variance of the estimation method. For a single model f_D we can interpret this equation by saying that a good method should be no biased and have as little variance as possible between different realizations.

If we rewrite the error decomposition in the form:

$$\mathrm{E}[(t - \mathrm{E}[f_D(\mathbf{x})|D])^2|D] = \mathrm{Bias}^2 + \sigma_\varepsilon^2 = \mathrm{MeanError} - \mathrm{Variance}, \tag{3}$$

we can reinterpret this equation in the following way: using the average $\mathrm{E}[f_D|D]$ as estimator, the generalization error can be reduced if we are able to produce fairly accurate models f_D (small MeanError[1]) while, at the same time, allowing them to produce the most diverse predictions at

[1] In this work we use the terms Accuracy and MeanError with the same meaning. In other cases Accuracy is defined as –MeanError or in another mathematical form that has a derivative opposite to MeanError's one.

every point (large Variance). Of course, there is a trade-off between these two conditions and several previous works [8, 14] discussed how to find a good compromise between mean error and diversity on ANN ensembles.

3 Learning in random subspaces

RF, as was discussed in the Introduction, is one of the most successful tree-based ensemble methods. RF combines two different sources of diversity: i) each tree in the ensemble is grown on a bootstrap sample of the original dataset and ii) when growing each tree, only a small random subset of features is considered at each node. By this second condition the learning algorithm is restricted to minimize its cost function only in randomly selected subspaces of the original hypothesis space. According to Breiman, this procedure produces less correlated trees while keeping a low mean error.

We can incorporate this last source of diversity to ANN ensembles with simple modifications of the training procedure. All practical ANN learning methods are based on an iterative minimization of a cost function over the vector space of possible weight values [1]. In all cases we can easily restrict the minimization procedure to random subspaces by the following procedure:
i) select at random a subset of weights.
ii) iterate a few cycles the learning algorithm but limiting it to change only the selected weights.
iii) iterate steps i) and ii) until a stopping criterion is reached.
Of course, different stopping criteria or random selection strategies can be implemented, leading to slightly different versions of the method. It worth mention that at step ii) all weights are available to the training algorithm to compute ANN outputs but it is limited to train (change) only the selected subset of weights.

This work is limited to model regression problems using ensembles of ANNs with a single hidden layer (with sigmoid activation functions) and a single output unit (with linear activation). For this particular setting we choose to keep all weights connecting the output unit to the hidden layer[2] and to limit the random selection to weights connecting the hidden layer to the input units using the following two strategies:
i) simply select at random a fraction F of all weights and pass them to the learning method. We call this strategy weight selection (WS),
ii) select a fraction F of input units and pass all weights coming from these units to the learning method. We call this strategy input selection (IS).
We also consider two different stopping criteria for the training of individual ANNs. In the first case, called optimal training (OT), we use the out-of-bag data as a validation set in order to monitor the performance of each ANN on unseen data and avoid overfitting. This is the most usual strategy for ANN or tree based ensembles. It is well known [3, 8] that some degree of overfitting of the individual members can be of benefit for the ensemble performance. We thus use a second criterion by which we train all ANNs a fixed number of epochs, long enough as to be sure that we overfit the training data. We call this second criterion full training (FT). Combining the two selection and stopping criteria plus different values of the fraction F of selected weights we produce diverse methods with different compromises between accuracy and variance.

[2] As all ANNs have only one output unit, selecting a subset of connections for that unit can reduce the effective complexity of the model and produce an accuracy decrease without the corresponding variance increase.

4 Experimental Settings

We evaluate the algorithms described in the previous section by applying them to several benchmark databases: the synthetic Friedman #1 and #2, the real-world Boston Housing and Ozone and two times series, Sunspots and Ikeda. In the following we give brief descriptions of the databases and details on the experimental settings.

4.1 Datasets

Friedman #1

The Friedman #1 synthetic data set corresponds to vectors with 10 input and one output variables generated according to

$$t = 10\sin(x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon,$$

where ε is Gaussian noise ($N(\mu = 0, \sigma = 1)$) and $x_1, \ldots x_{10}$ are uniformly distributed over the interval $[0, 1]$. Notice that $x_6, \ldots x_{10}$ do not enter in the definition of t and are only included to add input noise. We generate 1100 sample vectors and consider ANNs with 10:10:1 architectures.

Friedman #2

Friedman #2 has four independent variables and the target data are generated according to

$$y = x_1^2 + \sqrt{x_2 x_3 - (x_2 x_4)^{-2}} + \varepsilon,$$

where the zero-mean, normal noise is adjusted to give a noise-to-signal power ratio 1:3. The variables x_i are uniformly distributed in the ranges

$$0 < x_1 < 100, \quad 20 < \frac{x_2}{2\pi} < 280, \quad 0 < x_3 < 1, \quad 1 < x_4 < 11.$$

We generate 1100 patterns and consider 4:10:1 ANNs.

Boston Housing

This data set, from the UCI machine learning repository, consists of 506 training vectors with 11 input variables and one target output. The inputs are mainly socioeconomic information from census tracts on the greater Boston area and the output is the median housing price in the tract. We select an 11:8:1 architecture for ANNs.

Ozone

The Ozone data correspond to meteorological information (humidity, temperature, etc.) related to the maximum daily ozone at a location in Los Angeles area. Removing missing values one is left with 330 training vectors, containing 8 inputs and one target output in each one. The data set can be downloaded University of California at Berkeley. We select a 8:8:1 architecture in this case.

Method	Fried#1	Fried#2	Ozone	Boston	Ikeda	SSP
Bagging	0.30	0.12	0.25	0.12	0.29	0.12
WS-90%	0.89	-	1.02	0.99	0.99	-
WS-80%	0.94	1.03	0.99	1.02	1.00	-
WS-70%	0.97	1.02	1.04	1.04	1.07	1.03
WS-60%	1.00	1.02	1.01	0.97	1.04	1.03
WS-50%	1.03	1.04	1.02	0.95	1.05	1.02
WS-40%	-	-	-	-	-	1.03
IS-90%	0.99	-	-	0.94	1.00	-
IS-80%	0.98	1.01	1.00	0.98	1.00	-
IS-70%	0.97	1.00	1.02	0.91	0.97	0.99
IS-60%	0.93	0.99	0.98	0.98	1.02	0.99
IS-50%	0.90	0.99	0.98	1.05	1.04	0.99
IS-40%	-	-	-	-	-	0.98
IS-30%	-	-	-	-	-	0.97

Table 1: Experimental results for the Optimal Training stopping criterion. Bagging results are given in NMSE units. Results of all other methods are in relative units to the corresponding bagging value (i.e., a value lower than one means increasing in performance over bagging). Each row corresponds to a different selection strategy and fraction of selected weights.

Ikeda

The Ikeda laser map, which describes instabilities in the transmitted light by a ring cavity system, is given by the real part of the complex iterates

$$z_{n+1} = 1 + 0.9 z_n \exp\left[0.4i - \frac{6i}{(1 + |z_n|^2)}\right].$$

We generate 605 iterates and create 600 vectors using as inputs the last five values of the time series and as output the corresponding next value. We consider 5:8:1 ANNs.

Sunspots

The sunspots (SSP) time series is one of the most used benchmarks in time series prediction. It is the record of the yearly average of the number of sunspots (dark blotches on the sun mainly caused by magnetic activity) since 1700 to 1999. We generate 287 vectors using as inputs the last 12 values of the time series and as output the corresponding next value. We consider 12:6:1 ANNs for this problem.

4.2 Evaluation methods

For all six datasets we use ensembles with 50 ANNs. We selected this number of networks after checking in all cases that there are practically no performance improvements with bigger ensembles. Each ANN is trained with the standard back-propagation algorithm with momentum[1]. Training parameters (momentum, learning rate and # of epochs) were selected by internal cross validation but without a in-depth search for optimal values, because we are mainly interested in the relative performance of different ensemble methods.

Method	Fried#1	Fried#2	Ozone	Boston	Ikeda	Ssp
Bagg-FT	0.93	1.43	1.10	0.97	0.99	1.02
WS-70%	0.85	1.40	1.14	0.91	1.00	0.96
WS-60%	0.90	1.33	1.15	0.90	1.00	0.96
WS-50%	0.94	1.39	1.13	0.91	1.01	0.94
WS-40%	0.97	1.39	1.17	0.90	1.06	0.95
WS-30%	1.00	-	1.15	-	-	0.94
IS-70%	0.88	-	1.12	0.90	1.00	0.95
IS-60%	0.84	1.42	1.13	0.84	0.99	0.93
IS-50%	0.83	1.35	1.17	0.88	1.00	0.92
IS-40%	0.79	1.37	1.10	0.91	1.00	0.93
IS-30%	0.75	1.35	1.11	0.91	1.00	0.91

Table 2: Experimental results for the Full Training stopping criterion. All results are in relative units to the corresponding bagging value (see Table 1). Each row corresponds to a different selection strategy and fraction of selected weights.

All the results given in the next section are averages over 100 runs of each method. We repeated 10 times a 10-folds cross validation procedure, using alternatively one fold as test set and the remaining nine as training set. For all the methods under evaluation we use exactly the same 100 partitions in training and test sets.

The results quoted below are given in terms of the normalized mean-squared test error:

$$NMSE_T = \frac{MSE_T}{\sigma_D^2}, \tag{4}$$

defined as the mean-squared error on the test set T divided by the variance of the total data set D. For easy of interpretation, the results of the baseline method (bagging of optimally trained ANNs) are given in these units and the results of the other methods are given relative to these values. For example, the value 0.89 corresponding to Fried#1, WS-90% in Table 1 means that the WS-90% strategy gives more than a 10% decrease in test set prediction error over plain bagging.

5 Experimental results

We start our analysis evaluating the more typical OT stopping criterion. For both WS and IS selection strategies we use several values of the fraction F of selected weights in the 30–90% range. In Table 1 we show the corresponding results, including for comparison the results of bagging of optimally trained ANNs. Note that the only difference between bagging and the WS and IS strategies is the limitation on the last two methods to train only the selected subset of weights. For the WS strategy the results are poor. Only for Friedman#1 and Boston there are improvements over bagging for some values of F. In all other cases the OT–WS strategy produce worse results than bagging. On the other hand, the IS strategy outperforms WS in all six datasets. For this method, OT–IS, there are consistently better than bagging results for Friedman#1, Boston and SSP, and similar to bagging prediction errors for the other three

Dataset	Fried#1		Boston		Ssp	
Method	Acc	Var	Acc	Var	Acc	Var
Bagging	0.46	0.16	0.20	0.086	0.156	0.033
Bagg-FT	1.75	3.32	1.41	2.17	1.02	0.99
WS-70%	1.79	3.51	1.29	1.83	1.61	4.05
WS-60%	1.82	3.51	1.27	1.77	1.56	3.83
WS-50%	1.88	3.65	1.23	1.67	1.49	3.57
WS-40%	1.96	3.83	1.21	1.61	1.43	3.21
WS-30%	-	-	-	-	1.34	2.85
IS-70%	1.73	3.36	1.41	2.10	1.67	4.40
IS-60%	1.72	3.39	1.34	2.01	1.63	4.20
IS-50%	1.70	3.38	1.32	1.93	1.48	3.60
IS-40%	1.66	3.33	1.33	1.89	1.43	3.31
IS-30%	1.64	3.34	1.27	1.82	1.34	2.96

Table 3: Mean Error and Variance for three datasets (Fried#1, Boston and SSP) with similar behavior. Results are given in units relative to the corresponding bagging results. Columns labeled Acc show accuracy values (lower values are better) and columns labeled Var show variance values (bigger values are better).

problems. Analysing the dependence on the F value there is not a clear pattern, in some cases there are improvements using small subspaces and in other cases the opposite is true.

We repeat the analysis for the FT stopping criterion. In table 2 we show the new results, but including this time the results of bagging of fully trained ANNs (bagg-FT). Again, for the FT stopping criterion, the only difference between bagg-FT and the WS and IS strategies is that the last two methods train only a selected subset of weights. The bagg-FT results are similar to plain bagging for three datasets, worst for Fried#2 and Ozono and better only for Fried#1. IS is almost always better than WS, which is consistent with the OT results, suggesting that the IS strategy is more efficient. Comparing with bagg-FT there are two different behaviors. For Fried#1, Boston and SSP there are now clear improvements over bagging and bagg-FT. On the other side, for Fried#2, Ozono and Ikeda the results are similar or slightly worst than bagg-FT, and clearly worst than bagging for the first two. As in the OT case, there is not a clear pattern for the dependence on the F value, which seems to be problem-dependent.

Comparing the OT and FT stopping criteria, both show improvements over bagging on the same three datasets and some decrease in performance on the other three, but the FT strategy produces bigger differences with bagging in all cases.

5.1 Accuracy vs. Diversity

In order to gain some insight on the behavior of the new methods we also estimate the accuracy and diversity components of the prediction error according to section 2, equations 2 and 3. We limit the analysis to the IS strategy, which produces the bigger improvements over bagging.

In Tables 3 and 4 we present the corresponding results; for easier comparison, we again give them normalized by the mean accuracy and diversity of plain bagging and give bagging results in NMSE units. In Table 3 we present the results for the three datasets that clearly improve on

Dataset	Fried#2		Ozone		Ikeda	
Method	Acc	Var	Acc	Var	Acc	Var
Bagging	0.16	0.035	0.31	0.058	0.716	0.43
Bagg-FT	3.07	8.87	2.05	6.20	2.05	2.74
WS-70%	2.43	6.08	1.85	4.96	1.95	2.56
WS-60%	2.39	6.14	1.82	4.76	2.03	2.71
WS-50%	2.56	6.68	1.75	4.46	1.99	2.62
WS-40%	2.48	6.34	1.58	3.38	2.03	2.71
WS-30%	-	-	1.48	2.97	-	-
IS-70%	2.80	7.67	1.96	5.68	2.06	2.75
IS-60%	2.34	5.80	1.89	5.20	2.08	2.78
IS-50%	2.33	5.69	1.81	4.63	2.00	2.65
IS-40%	2.33	5.82	1.79	4.79	1.99	2.63
IS-30%	-	-	1.67	4.10	2.00	2.66

Table 4: Mean Error and Variance for three datasets (Fried#2, Ozone and Ikeda) with similar behavior. Results are given in units relative to the corresponding bagging results. Columns labeled Acc show accuracy values (lower values are better) and columns labeled Var show variance values (bigger values are better).

bagging, Fried#1, Boston and SSP, and left the other three datasets for Table 4. The bagg-FT values in both tables show that the FT strategy produces a lost in accuracy linked with an increase in variance, relative to bagging values, in all but the SSP dataset. Comparing both groups of datasets, the first one (Table 3) has lower increases in both accuracy and variance over bagging.

The accuracy and variance values for the WS and IS strategies are not easy to analyze. For SSP there is, as expected, a four times increase in variance coupled with a moderate loss in accuracy (bigger Acc values), given as result a reduced prediction error (see Table 2). But the results for Fried#1 and Boston are unexpected. In both datasets the better performance is associated to accuracy values better than bagg-FT together with increases in variance similar to bagg-FT. For all datasets in Table 4 our subspace strategies also produce better than bagg-FT accuracies, but coupled in this case with a reduction in variance, which seems to be the cause of the lost in prediction capabilities.

5.2 Adding noisy inputs

The three datasets that show improvements in prediction error over bagging and bagg–FT have some very noisy or irrelevant inputs. The Fried#1 datasets has 5 white noise inputs. Our embedding of the SSP datasets uses the last 12 values of the series, but is know that a non-uniform embedding with 3 values gives optimal results. The inputs in the Boston problem consist of very noisy socioeconomic information from census tracts. The other three datasets have fewer inputs and all relevant. To check if this can be the origin of the different behavior of the two groups we conducted a small experiment adding white-noise inputs to the Fried#2 dataset. We produced two new datasets, one with 5 added noisy inputs and a second one with 15 more noisy inputs. We show the new results in Table 5. The addition of useless inputs produces a better performance of our FT subspace methods relative to that of bagging and bagg–FT. The

	Plain			+5 Features			+20 Features		
Method	Err	Acc	Var	Err	Acc	Var	Err	Acc	Var
Bagging	0.12	0.16	0.035	0.19	0.60	0.41	0.27	0.69	0.42
Bagg-FT	1.43	3.08	8.87	0.96	1.00	1.01	0.99	0.97	0.96
WS-FT-50%	1.39	2.56	6.68	0.98	1.02	1.02	1.04	1.04	1.03
IS-FT-50%	1.35	2.33	5.69	1.03	1.25	1.34	0.98	0.99	0.99

Table 5: Prediction error (Err), Accuracy (Acc) and variance (Var) for the Friedman#2 dataset with added white-noise input features. Results are given in units relative to the corresponding bagging results.

results are equivalent for 5 or 20 added features, but for 20 features the IS-FT strategy gives the best result of all methods.

5.3 Comparison with other methods

As a final investigation on our random subspace methods we compare them with three other state-of-the-art regression methods. We selected two tree-based ensemble methods, bagging and Random Forest [2, 3] and Support Vector Machines (SVM) [15] with a gaussian kernel. We selected the FT-IS-50% method as a good representative of our new methods. To have a fair comparison we use exactly the same 100 partitions in train/test set. We use 1000 trees for both Bagging and RF and set all other parameters to the default values given by Breiman. For SVM we selected the C and γ parameters using internal cross-validation on each train set.

The corresponding results are shown in Table 6. On Fried#1, Boston, Ikeda and SSP datasets the FT–IS–50% method gives the best results. On the other two datasets, tree-bagging wins in one case and RF in the other. But in almost all cases tree-bagging, RF and SVM have bigger than one results, showing that in fact it is really difficult to improve over the results of plain bagging of optimally trained ANNs.

6 Conclusions

In this work we introduced new methods for neural network ensemble construction that follow Random-Forest-like strategies in order to increase the diversity of the members. We selected two strategies for the random selection of weights to be trained, the WS strategy that simply selects

Method	Fried#1	Fried#2	Ozone	Boston	Ikeda	Ssp
Bagging (trees)	1.13	1.22	1.09	1.11	1.47	1.66
RF (trees)	1.26	2.53	1.00	1.09	1.66	2.02
SVM (gaussian)	0.85	1.39	1.14	1.94	1.21	1.32
FT-IS-50%	0.83	1.35	1.17	0.88	1.00	0.92

Table 6: Prediction error comparison with other state-of-the-art methods. Results are given in units relative to the corresponding bagging results.

at random a fraction F of all weights and the IS strategy that selects a fraction F of input units and all the weights coming from these units. We also consider two different stopping criteria for the training of individual ANNs, the typical optimal training (OT) and the full training (FT) by which we train all ANNs a fixed number of epochs, long enough as to overfit the training data. We evaluated the combination of the two selection and stopping criteria plus different values of the fraction F of selected weights on six real-world and artificial regression problems. We found that the IS strategy usually outperforms the WS one, and that the FT stopping method gives better than OT results for three datasets where all subspace strategies outperform bagging. On the other three datasets both stopping criteria works no better than bagging and in particular FT gives the worst results.

The analysis of the accuracy and variance values suggest that in most cases the subspace strategies do not produce an increase in variance over fully trained bagging ensembles. Instead, they seem to produce better prediction errors by a combination of an increased accuracy with similar variance than bagging. This result is very interesting and requires further investigations.

We have also evaluated the addition of noisy inputs to the Fried#2 dataset. The results of that experiment supports the idea that our subspace methods perform better in problems with several noisy or irrelevant inputs. This fact suggests an important application field, the calibration of spectrometric instrument in chemometrics.

Finally, we have also performed a comparison with other three state-of-the-art regression methods. The FT-IS strategy was the only method of the four evaluated capable of clearly outperforming bagging of ANNs in several datasets.

As future work we are also considering extending the proposed methods to classification problems and to evaluate other random selection strategies.

Acknowledgments

We acknowledge support for this project from the ANPCyT (grant PICT 2003 11-15132).

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, London, 1995.

[2] L. Breiman. Bagging predictors. *Machine Learning* 24:123-140, 1996.

[3] L. Breiman. Random forests. *Machine Learning* 45:5-32, 2001.

[4] L. Breiman. Randomizing outputs to increase prediction accuracy. *Machine Learning* 40:229-242, 2000.

[5] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23-37, Springer Verlag, 1995.

[6] S. Geman, E. Bienenstock and R. Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation* 4:1-58, 1992.

[7] P. Geurts, D. Ernst and L. Wehenkel. Extremely randomized trees. *Machine Learning* 63:342, 2006

[8] P. M. Granitto, P. F. Verdes and H. A. Ceccatto. Neural Networks Ensembles: Evaluation of Aggregation algorithms. *Artificial Intelligence* 163:139-162, 2005.

[9] P. M. Granitto, P. F. Verdes, H. D. Navone and H. A. Ceccatto. A Late-stopping Method for Optimal Aggregation of Neural Networks. *Internation Journal of Neural Networks* 11:305-310, 2001.

[10] L. I. Kuncheva *Combining Pattern Classifiers*. Wiley-Interscience, New Jersey, 2004.

[11] G. Martinez-Muñoz, A. Sanchez-Martinez, D. Hernandez-Lobato and A. Suarez. Building Ensembles of Neural Networks with Class-Switching. Proceedings of the ICANN 2006, Part I, LNCS 4131, 178187, 2006.

[12] J J Rodriguez, L. I. Kuncheva and C. J. Alonso. Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1619-1630, 2006.

[13] D. Opitz and R. Maclin. Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, 11:169-198, 1999.

[14] B. Rosen. Ensemble learning using decorrelated neural networks. *Connection Science*. Special Issue on Combining Artificial Neural Nets: Ensemble Approaches 8(3&4):373-384, 1996.

[15] B. Scholkopf and A. J. Smola. *Learning with Kernels* MIT Press, Cambridge, 2002.

[16] H. Schwenk and Y. Bengio. Boosting neural networks. *Neural Computation* 12:1869-1887, 2000.

[17] A. J. C. Sharkey, editor. *Combining Artificial Neural Nets*. Springer-Verlag, London, 1999.

Acceptability Semantics and Contextual Defeat Relations in Extended Frameworks

Diego C. Martínez Alejandro J. García Guillermo R. Simari
Comisión de Investigaciones Científicas y Técnicas - CONICET
Artificial Intelligence Research and Development Laboratory
Department of Computer Science and Engineering
UNIVERSIDAD NACIONAL DEL SUR
Bahía Blanca - Buenos Aires - República Argentina
E-mail: {dcm,ajg,grs}@cs.uns.edu.ar

Abstract

In this work, contexts for extended argumentation frameworks (EAF) are defined. A context for an EAF is another framework where original arguments, conflicts and preferences are kept, while introducing new arguments leading to new defeat relations. Thus, the context may interfere with the original classification of arguments, inducing new set of extensions. These semantic change in the outcome of an extended framework in a particular context is characterized, and Dung's acceptability concept is analyzed on this basis.

1 Introduction

Argumentation has became an important subject of research in Artificial Intelligence and it is also of interest in several disciplines, such as Logic, Philosophy and Communication Theory. This wide range of attention is due to the constant presence of argumentation in many activities, most of them related to social interactions between humans, as in civil debates, legal reasoning or every day dialogues. Basically, an argument is a piece of reasoning that supports a claim from certain evidence. The tenability of this claim must be confirmed by analyzing other arguments for and against such a claim. In formal systems of defeasible argumentation, a claim will be accepted if there exists an argument that supports it, and this argument is acceptable according to an analysis between it and its counterarguments. After this dialectical analysis is performed over the set of arguments in the system, some of them will be *acceptable*, *justified* or *warranted* arguments, while others will be not. The study of the acceptability of arguments is the search for rationally based positions of acceptance in a given scenario of arguments and their relationships. It is one of the main concerns in Argumentation Theory.

Abstract argumentation systems [5, 12, 6, 1, 2] are formalisms for argumentation where some components remain unspecified, being the structure of an argument the main abstraction. In this kind of system, the emphasis is put on the semantic notion of finding

the set of accepted arguments. Most of them are based on the single abstract concept of *attack* represented as an abstract relation, and extensions are defined as sets of possibly accepted arguments. For two arguments $\mathcal{A}$ and $\mathcal{B}$, if $(\mathcal{A}, \mathcal{B})$ is in the attack relation, then the acceptance of $\mathcal{B}$ is conditioned by the acceptance of $\mathcal{A}$, but not the other way around. It is said that argument $\mathcal{A}$ attacks $\mathcal{B}$, and it implies a priority between conflicting arguments. It is widely understood that this priority is related to the argument strengths. Several frameworks do include an argument order [1, 3, 4], although this order is used at another level, as the classic attack relation is kept.

In [8, 7] an extended abstract argumentation framework (EAF)is introduced, where two kinds of defeat relations are present. These relations are obtained by applying a preference criterion between conflictive arguments. The conflict relation is kept in its most basic, abstract form: two arguments are in conflict simply if both arguments cannot be accepted simultaneously. The preference criterion subsumes any evaluation on arguments and it is used to determine the direction of the attack. This argument comparison, however, is not always succesful and therefore attacks, as known in classic frameworks, are no longer valid.

An argumentation framework Φ is basically the modelization of a knowledge base conformed by arguments. These arguments interact each other and then several possible outcomes as sets of accepted arguments are obtained. However, it is possible for this outcome to be different when new arguments are taken into account. These new arguments are considered the *context* of the framework Φ. For example, when a person is judged in a regular trial, several arguments for and against its innocence are exposed by the district attorney and by the defender lawyer. This set of arguments, say $Case$, is about the assumptions and facts of the particular case. Another set of arguments, however, is taked into account: those produced by the juror and the judge. Thus, the set $Case$ is placed in a special *context*: the actual trial. If the person is declared guilty, its lawyers may appeal to an upper level of Justice Court. Basically, they want to expose its arguments in a *different context*, in order to plead the defended not guilty.

We think situations like above may be modeled using extended abstract frameworks. This paper is organized as follows. In Section 2 our extended argumentation framework is presented. In Section 3 the notion of *contexts* for EAF is introduced. In Section 4 the behaviour of contexts is analyzed according to Dung's acceptability semantics [5]. Finally, the conclusions and future work are presented in Section 5.

2 Extended Argumentation Framework

In our extended argumentation framework three relations are considered: *conflict*, *subargument* and *preference* between arguments. The definition follows:

Definition 1.
An extended abstract argumentation framework (EAF) is a quartet Φ=$\langle AR, \sqsubseteq, \mathbf{C}, \mathbf{R}\rangle$*, where* AR *is a finite set of arguments, and* $\sqsubseteq$, $\mathbf{C}$ *and* $\mathbf{R}$ *are binary relations over* AR *denoting respectivelly subarguments, conflicts and preferences between arguments.*

Arguments are abstract entities, as in [5], that will be denoted using calligraphic uppercase letters, possibly with indexes. In this work, the subargument relation is not relevant for the topic addressed. Basically, it is used to model the fact that arguments may

include inner pieces of reasoning that can be considered arguments by itself, and it is of special interest in dialectical studies [9]. Hence, unless explicity specified, in the rest of the paper $\sqsubseteq = \emptyset$. The conflict relation $\mathbf{C}$ states the incompatibility of acceptance between arguments. Given a set of arguments S, an argument $\mathcal{A} \in S$ is said to be in conflict in S if there is an argument $\mathcal{B} \in S$ such that $\{\mathcal{A}, \mathcal{B}\} \in \mathbf{C}$. The relation $\mathbf{R}$ is introduced in the framework and it will be used to evaluate arguments, modelling a preference criterion based on a measure of strength.

Definition 2.
Given a set of arguments AR, an argument comparison criterion $\mathbf{R}$ *is a binary relation on AR. If $\mathcal{A}\mathbf{R}\mathcal{B}$ but not $\mathcal{B}\mathbf{R}\mathcal{A}$ then $\mathcal{A}$ is strictly preferred to $\mathcal{B}$, denoted $\mathcal{A} \succ \mathcal{B}$. If $\mathcal{A}\mathbf{R}\mathcal{B}$ and $\mathcal{B}\mathbf{R}\mathcal{A}$ then $\mathcal{A}$ and $\mathcal{B}$ are indifferent arguments with equal relative preference, denoted $\mathcal{A} \equiv \mathcal{B}$. If neither $\mathcal{A}\mathbf{R}\mathcal{B}$ or $\mathcal{B}\mathbf{R}\mathcal{A}$ then $\mathcal{A}$ and $\mathcal{B}$ are incomparable arguments, denoted $\mathcal{A} \bowtie \mathcal{B}$.*

For two arguments $\mathcal{A}$ and $\mathcal{B}$ in AR, such that the pair $\{\mathcal{A}, \mathcal{B}\}$ belongs to $\mathbf{C}$ the relation $\mathbf{R}$ is considered. In order to elucidate conflicts, the participant arguments must be compared. Depending on the preference order, two notions of argument defeat are derived.

Definition 3.
Let Φ=$\langle AR, \sqsubseteq, \mathbf{C}, \mathbf{R}\rangle$ be an EAF and let $\mathcal{A}$ and $\mathcal{B}$ be two arguments such that $(\mathcal{A}, \mathcal{B}) \in \mathbf{C}$. If $\mathcal{A} \succ \mathcal{B}$ then it is said that $\mathcal{A}$ is a proper defeater of $\mathcal{B}$. If $\mathcal{A} \equiv \mathcal{B}$ or $\mathcal{A} \bowtie \mathcal{B}$, it is said that $\mathcal{A}$ is a blocking defeater of $\mathcal{B}$, and viceversa. An argument $\mathcal{B}$ is said to be a defeater of an argument $\mathcal{A}$ if $\mathcal{B}$ is a blocking or a proper defeater of $\mathcal{A}$.

Example 1. *Let $\Phi_1 = \langle AR, \sqsubseteq, \mathbf{C}, \mathbf{R}\rangle$ be an EAF where $AR = \{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}\}$, $\sqsubseteq = \emptyset$, $\mathbf{C} = \{\{\mathcal{A}, \mathcal{B}\}, \{\mathcal{B}, \mathcal{C}\}, \{\mathcal{C}, \mathcal{D}\}\}, \{\mathcal{C}, \mathcal{E}\}\}$ and $\mathcal{A} \succ \mathcal{B}, \mathcal{B} \succ \mathcal{C}$, $\mathcal{E} \bowtie \mathcal{C}$, $\mathcal{C} \equiv \mathcal{D}$.*

Extended abstract frameworks can also be depicted as graphs, with different types of arcs, called *EAF-graphs*. We use to represent arguments as black triangles. An arrow ($\rightarrow$) is used to denote proper defeaters. A double-pointed straight arrow ($\leftarrow\!\!\rightarrow$) connects blocking defeaters considered equivalent in strength, and a double-pointed zig-zag arrow ($\leftarrow\!\!\sim\!\!\rightarrow$) connects incomparable blocking defeaters. In Figure 1, the framework Φ_1 is shown. Argument $\mathcal{A}$ is a proper defeater of $\mathcal{B}$. Argument $\mathcal{B}$ is a proper defeater of $\mathcal{C}$, and $\mathcal{E}$ is an incomparable blocking defeater of $\mathcal{C}$ and viceversa. Argument $\mathcal{D}$ and $\mathcal{C}$ are blocking defeaters being equivalent in strength.

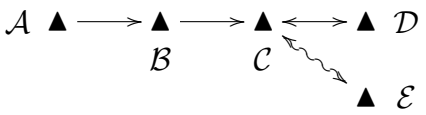


Figure 1: EAF-graph of framework Φ_1

In the next section we formally present *contexts* for extended argumentation frameworks and several semantic notions around this concept.

3 Contexts

An extended argumentation framework may be considered in different contexts, where its elements are still valid and well-defined, but interacting with new arguments. This is formalized as follows.

Definition 4.
Let Φ=$\langle AR_1, \sqsubseteq_1, \mathbf{C}_1, \mathbf{R}_1\rangle$ *be an extended argumentation framework. A context for* Φ *is a tuple* $\langle AR_2, \sqsubseteq_2, \mathbf{C}_2, \mathbf{R}_2\rangle$ *such that*

- $AR_1 \subseteq AR_2$,
- *For any pair of conflicting arguments* $(\mathcal{A}, \mathcal{B}) \in \mathbf{C_2}$ *such that* $\mathcal{A}, \mathcal{B} \in AR_1$ *then* $(\mathcal{A}, \mathcal{B}) \in \mathbf{C_1}$.
- *If* $\mathcal{A}\mathbf{R}_2\mathcal{B}$ *for any pair of arguments* $\mathcal{A}, \mathcal{B} \in AR_1$, *then* $\mathcal{A}\mathbf{R}_1\mathcal{B}$.
- *For any arguments* $\mathcal{X}, \mathcal{Y} \in AR_2$ *such that* $\mathcal{X} \sqsubseteq_2 \mathcal{Y}$, *if* $\mathcal{X} \in AR_2 - AR_1$ *then* $\mathcal{Y} \in AR_2 - AR_1$.

Arguments in $AR_2 - AR_1$ *are called contextual arguments.*

Definition 4 states that a context for an extended argumentation framework Φ_1 is just another extended framework Φ_2 where

1. all of the arguments in Φ_1 are included in Φ_2, and
2. no conflict between arguments of Φ_1 is added by Φ_2, and
3. preferences established in Φ_1 remain intact in Φ_2, and
4. Φ_2 may include new superarguments of arguments in Φ_1, but not the other way around.

Any framework is said to be a context for itself or a *self-context*.

In Figure 2 the general idea of framework context is shown. New arguments are present, which are able to defeat or to be defeated by arguments in Φ_1. Note that if Φ_2 is a context for Φ_1 then Φ_1 is a restriction of Φ_2, in the sense of [10], taking into account subarguments. That is, the set AR_1 must be *structurally complete*: it includes the subarguments of all of its elements.

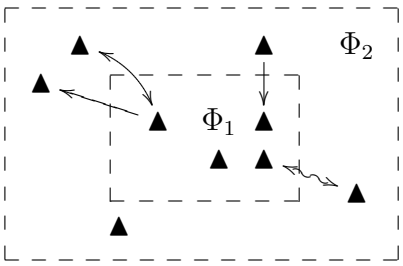


Figure 2: EAF Φ_2 is a context for EAF Φ_1

Example 2. *Let* $\Phi_1 = \langle AR_1, \sqsubseteq_1, \mathbf{C}_1, \mathbf{R}_1 \rangle$ *be an extended abstract framework where* $AR_1 = \{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$, $\mathbf{C_1} = \{\{\mathcal{A}, \mathcal{B}\}, \{\mathcal{B}, \mathcal{C}\}\}$, $\mathcal{A} \succ \mathcal{B}$ *and* $\mathcal{B} \bowtie \mathcal{C}$. *The following framework* $\Phi_2 = \langle AR_2, \sqsubseteq_2, \mathbf{C}_2, \mathbf{R}_2 \rangle$ *is a context for* Φ_1, *where* $AR_2 = \{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}\}$, $\mathbf{C_2} = \{\{\mathcal{A}, \mathcal{B}\}, \{\mathcal{B}, \mathcal{C}\}\{\mathcal{C}, \mathcal{D}\}, \{\mathcal{A}, \mathcal{E}\}\}$, $\mathcal{A} \succ \mathcal{B}$, $\mathcal{B} \bowtie \mathcal{C}$, $\mathcal{E} \succ \mathcal{A}$ *and* $\mathcal{D} \equiv \mathcal{C}$, *is a context for* Φ_1.

The context Φ_X for an extended framework Φ may also be placed in a new context Φ_Y. Even more, this new context Φ_Y is also a context for Φ.

Proposition 1. *Let* Φ_1 *and* Φ_2 *be two extended argumentation frameworks such that* Φ_2 *is a context for* Φ_1. *Then every extended framework* Φ_3 *such that* Φ_3 *is a context for* Φ_2, *it is also a context for* Φ_1.

Proof. Obvious from the definition. □

In order to evaluate the outcome of an argumentation framework in a particular context, we use the notation adopted by Baroni & Giacomin in [11], where semantic extensions are studied.

Definition 5. *[11]*
Given a generic argumentation semantic $\mathcal{S}$, *the set of extensions prescribed by* $\mathcal{S}$ *for an* EAF Φ *is denoted as* $\mathcal{E}_S(\Phi)$

The set of argument extensions induced by an EAF may change when its arguments are challenged by new arguments in the context. It is possible that an argument is no longer present in every extension, or to be included in a new one. Even more, an entire extension may not be valid in a specific context. The following definition introduces new terminology in relevant cases.

Definition 6.
Let $\Phi_1 = \langle AR_1, \sqsubseteq_1, \mathbf{C}_1, \mathbf{R}_1 \rangle$ *be an extended argumentation framework and let* $\Phi_2 = \langle AR_2, \sqsubseteq_2, \mathbf{C}_2, \mathbf{R}_2 \rangle$ *be a context for* Φ_1. *Let* $\mathcal{S}$ *be an argumentation semantic.*

- Φ_2 *is said to* $\mathcal{S}$*-confirm* Φ_1 *if* $\mathcal{E}_S(\Phi_1) = \mathcal{E}_S(\Phi_2)$.
- Φ_2 *is said to* $\mathcal{S}$*-preserve* Φ_1 *if for every extension* $X \in \mathcal{E}_S(\Phi_1)$, *there is an extension* $Y \in \mathcal{E}_S(\Phi_2)$ *such that* $X \subset Y$. *Every argument in* X *is said to be preserved by* Φ_2.
- Φ_2 *is said to* $\mathcal{S}$*-expand* Φ_1 *if* Φ_2 $\mathcal{S}$*-preserve* Φ_1 *and every extension* $Y \in \mathcal{E}_S(\Phi_2)$ *is a superset of an extension* X *in* $\mathcal{E}_S(\Phi_1)$.
- Φ_2 *is said* $\mathcal{S}$*-revise* Φ_1 *if exists an extension* X *in* $\mathcal{E}_S(\Phi_1)$ *such that no extension in* $\mathcal{E}_S(\Phi_2)$ *is a superset of* X. *The set* X, *as an extension, is said to be revised by* Φ_2. *Also it is said that* Φ_2 *revises* Φ_1 *in* X.

The following table summarizes the concepts presented in Definition 6, and captures the essential meaning of every case.

Concept	Meaning
$\mathcal{S}$-confirm	*No extension is changed or added*
$\mathcal{S}$-preserve	*The same alternatives of acceptance are available, but some extensions of Φ_2 may propose new sets of arguments for acceptance.*
$\mathcal{S}$-expand	*There is always an extension of Φ_2 that includes a valid alternative of acceptance for Φ_1 according to $\mathcal{S}$.*
$\mathcal{S}$-revise	*The alternative X of acceptance in Φ_1 is no longer valid in Φ_2 as a whole, $i.e.$ the extension is "broken" or discarded by Φ_2.*

It is clear that any EAF Φ $\mathcal{S}$-confirms Φ. When a context Φ_X $\mathcal{S}$-confirm an EAF Φ_Y then every argument in Φ_X (if any) is defeated by at least an argument in an extension of Φ_Y. Simple frameworks and contexts exhibiting these properties are shown in Example 3 and Figure 3.

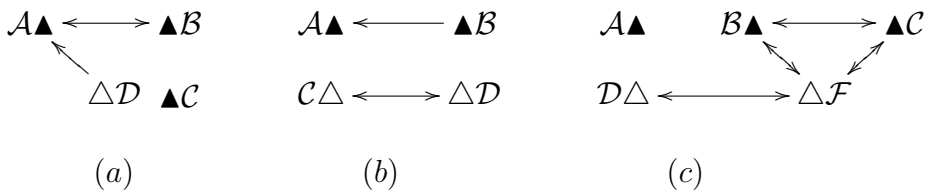


Figure 3: Frameworks and contexts

Example 3. *Consider the three frameworks depicted in Figure 2. In each case, arguments in Φ_1 are shown as black triangles and arguments in context Φ_2 are shown as white triangles. Let $\mathcal{P}$ be the admissibility-based preferred semantics and let $\mathcal{E}_{\mathcal{P}(\Phi)}$ the set of all preferred extensions of framework Φ. In the following table the preferred extensions and context properties are shown:*

Example	$\mathcal{E}_{\mathcal{P}(\Phi_1)}$	$\mathcal{E}_{\mathcal{P}(\Phi_2)}$	*Properties of Φ_2 with respect to Φ_1*
(a)	$\{\{\mathcal{A},\mathcal{C}\}, \{\mathcal{B},\mathcal{C}\}\}$	$\{$ $\{\mathcal{D},\mathcal{B},\mathcal{C}\}$ $\}$	*$\mathcal{P}$-revise, not $\mathcal{P}$-preserve, not $\mathcal{P}$-expand*
(b)	$\{\{\mathcal{B}\}\}$	$\{$ $\{\mathcal{B},\mathcal{C}\},\{\mathcal{B},\mathcal{D}\}$ $\}$	*$\mathcal{P}$-preserve, $\mathcal{P}$-expand, not $\mathcal{P}$-revise*
(c)	$\{\{\mathcal{A},\mathcal{B}\}, \{\mathcal{A},\mathcal{C}\}\}$	$\{$ $\{\mathcal{A},\mathcal{B},\mathcal{C}\}$, $\{\mathcal{A},\mathcal{C},\mathcal{D}\}$, $\{\mathcal{A},\mathcal{F}\}$ $\}$	*$\mathcal{P}$-preserve, not $\mathcal{P}$-expand, not $\mathcal{P}$-revise*

In the next section the acceptability semantics defined in [5] is analyzed for extended abstract frameworks and its contexts.

4 Contexts and acceptability-based semantics

The argumentation framework defined by Dung in [5] is the core of argument basic semantic notions. Its framework only includes arguments and attacks as a binary relation on arguments, the basic elements for semantics elaborations. The main contribution is the formalization of several argument extensions capturing rational sets of acceptance. The key notion is *acceptability of arguments*, defined here for extended abstract frameworks

Definition 7.
Let $\langle AR, \sqsubseteq, \mathbf{C}, \mathbf{R}\rangle$ be an EAF. An argument $\mathcal{A} \in AR$ is acceptable with respect to a set of arguments $S \subseteq AR$ if and only if every argument $\mathcal{B}$ defeating $\mathcal{A}$ is defeated by an argument in S.

Defeaters mentioned in Definition 7 may be either proper or blocking ones. It is also said that S is *defending* $\mathcal{A}$ against its attackers. The defense or reinstatement of arguments is a central concept on argumentation. Extensions are required to be free of inner conflicts, and thus the following definition is needed.

Definition 8.
A set of arguments $S \subseteq AR$ is said to be conflict-free if for all $\mathcal{A}, \mathcal{B} \in S$ it is not the case that $\{\mathcal{A}, \mathcal{B}\} \in \mathbf{C}$.

As said before, in Dung's approach several semantic notions are defined as argument extensions leading to rational positions of acceptance. These extensions can also be applied to extended frameworks and are summarized in the following definition.

Definition 9.
A set of arguments S is said to be

- *admissible if it is conflict-free and defends all its elements.*
- *a preferred extension if S is a maximal (for set inclusion) admissible set.*
- *a complete extension if S is admissible and it includes every acceptable argument w.r.t. S.*
- *a grounded extension if and only if it is the least (for set inclusion) complete extension.*
- *a stable extension if S is conflict-free and it attacks each argument not in S.*

The grounded extension of a framework Φ, denoted GE_Φ, is also the least fixpoint of a simple monotonic *characteristic* function:

$$F_{AF}(S) = \{\mathcal{A} : \mathcal{A} \text{ is acceptable wrt } S\}.$$

Several modifications to the classic Dung's framework are proposed in the literature, and new semantic notions were introduced. For example, in [6] the original framework is kept, while presenting a new argument extension. In [1], preferences between arguments are added to the framework and new semantic consideratios are made.

The following proposition uses Definition 6 applied to preferred and grounded extensions.

Proposition 2. *Let* $\Phi_1 = \langle AR_1, \sqsubseteq_1, \mathbf{C}_1, \mathbf{R}_1 \rangle$ *be an extended argumentation framework and let* $\Phi_2 = \langle AR_2, \sqsubseteq_2, \mathbf{C}_2, \mathbf{R}_2 \rangle$ *be a context for* Φ_1*. Let* $\mathcal{P}$ *and* $\mathcal{G}$ *be the preferred and grounded semantics respectively.*

- *If* Φ_2 $\mathcal{P}$*-preserves* Φ_1 *then every argument in an extension* $X \in \mathcal{E}_{\mathcal{P}}(\Phi_1)$ *is acceptable with respect to* AR_2*.*
- *If* Φ_2 $\mathcal{G}$*-preserves* Φ_1 *then also* Φ_2 $\mathcal{G}$*-expands* Φ_1*.*

Proof. If Φ_2 $\mathcal{P}$-preserves Φ_1 then every argument included in a preferred extension X of Φ_1 is also included in a preferred extension of Φ_2 and therefore is acceptable with respect to a set in AR_2. The grounded extension is unique (being a skeptical notion), and thus if Φ_2 $\mathcal{G}$-preserves Φ_1, then the grounded extension of Φ_2 includes every argument in the grounded extension of Φ_1. As these are the only sets in $\mathcal{E}_{\mathcal{P}}(\Phi_1)$ and $\mathcal{E}_{\mathcal{P}}(\Phi_2)$, then Φ_2 $\mathcal{G}$-expands Φ_1. □

As stated in Proposition 1, a context Φ_3 for a framework Φ_2 being a context for Φ_1, is in turn a context for Φ_1. As Φ_2 and Φ_3 are taking into account new arguments with respect to Φ_1, the extensions may vary among these frameworks. For a semantic notion $\mathcal{S}$, an argument $\mathcal{A}$ may be in an extension X_1 of $\mathcal{E}_{\mathcal{S}}(\Phi_1)$, but not in any extension of $\mathcal{E}_{\mathcal{S}}(\Phi_2)$. Later on, it is possible for $\mathcal{A}$ to be included in an extension of $\mathcal{E}_{\mathcal{S}}(\Phi_3)$, resembling argument reinstatement. The following proposition relates this situation in the particular case of acceptability semantics.

Proposition 3. *Let* Φ_1 *and* Φ_2 *be two extended argumentation frameworks such that* Φ_2 *is a context for* Φ_1 *and let* $\mathcal{G}$ *be the grounded extension semantics. If* Φ_2 $\mathcal{G}$*-revise* Φ_1*, it is possible to construct a context* Φ_3 *for* Φ_2 *such that* Φ_3 $\mathcal{G}$*-expand* Φ_1*.*

Proof. If Φ_2 $\mathcal{G}$-revise Φ_1, then a subset $S \subseteq GE_{\Phi_1}$ is not included in GE_{Φ_2}, due to new defeaters in $AR_2 - AR_1$. Let $S' = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_n, \}$ be the set of these defeaters. The extended framework $\Phi_3 = \langle AR_3, \sqsubseteq_3, \mathbf{C}_3, \mathbf{R}_3 \rangle$ is constructed as following:

- $AR_3 = AR_2 \cup \{\mathcal{Z}_1, \mathcal{Z}_2, ..., \mathcal{Z}_n\}$ where $\mathcal{Z}_i$, $1 \leq i \leq n$ is a new argument not appearing in AR_2.
- $\sqsubseteq_3 = \emptyset$,
- $\mathbf{C_3} = \mathbf{C_2} \cup \bigcup_i \{\mathcal{D}_i, \mathcal{Z}_i\}$
- $\mathcal{Z}_i \succ \mathcal{D}_i$ for all $\mathcal{D}_i, \mathcal{Z}_i$

As any argument $\mathcal{D}_i$ is defeated by a defeater-free argument ($\mathcal{Z}_i$), any threat over S introduced by Φ_2 is no longer valid, and then every argument in S is included in a grounded extension again. Also every $\mathcal{Z}_i$ is in the grounded extension as they are defeater-free arguments. □

4.1 Non-relevant arguments

Some arguments in a context may not be relevant for changes in semantic extensions. Basically, these are arguments such that its inclusion is not a threat for other arguments. Of course, this notion is considered under a particular semantic notion $\mathcal{S}$. In the following definition, non-relevant arguments are presented according to the classical grounded extension.

Definition 10.
Let $\Phi_1 = \langle AR_1, \sqsubseteq_1, \mathbf{C}_1, \mathbf{R}_1 \rangle$ *be an extended argumentation framework and let* $\Phi_2 = \langle AR_2, \sqsubseteq_2, \mathbf{C}_2, \mathbf{R}_2 \rangle$ *be a context for* Φ_1*. Let* $\mathcal{A}$ *be an argument in the grounded extension* GE_{Φ_1}*. A contextual argument* $\mathcal{X}$ *is said to be non-relevant for* $\mathcal{A}$ *if*

- $\mathcal{X}$ *does not directly nor indirectly defeats* $\mathcal{A}$*, or*
- *whenever* $\mathcal{X}$ *directly or indirectly defeats* $\mathcal{A}$*, it is directly or indirectly defeated by an argument in* GE_{Φ}

Non relevant arguments for $\mathcal{A}$ are those contextual arguments not being able to avoid the inclusion of $\mathcal{A}$ in the grounded extension of the context. This is important en several scenarios. Following the introductory analogy if Justice trials, non-relevant arguments are the main target of lawyers. These arguments may be viewed as a useless argument used by a member of the juror. It is useless because, even when defeating an argument in the case, it is already defeated by an argument in that case. These arguments are important in different ways. For example, a defender lawyer may want to introduce enough arguments to defeat any contextual argument defeating an argument exposed by himself. He is trying to maximize the number of non-relevant contextual arguments in that sense. On the other hand, he also wants to avoid the defeat of juror's arguments defeating arguments exposed by the District attorney. In this sense, he is trying to minimize the number of non-relevant arguments. Of course, they do not know *a priori* any of the contextual arguments. All they can do is to produce a set of arguments good enough to face any court.

5 Conclusions

An argumentation framework Φ is basically the model of a knowledge base based on arguments. These arguments interact each other and then several possible outcomes, as sets of accepted arguments, are obtained. However, it is possible for this outcome to be different when new arguments are taken into account. These new arguments are considered the *context* of the framework Φ. In this work, we formally defined *contexts* for extended argumentation frameworks (EAF). In general terms, a context for an EAF is another framework where original arguments, conflicts and preferences are kept, while new arguments are introduced, possibly leading to new defeat relations. As new argument interactions are present, the context may apply changes in the original classification of arguments, inducing new set of extensions. These semantic change in the outcome of an extended framework in a particular context was characterized, and Dung's acceptability concept was analyzed on this basis.

References

[1] Leila Amgoud and Claudette Cayrol. On the acceptability of arguments in preference-based argumentation. In *14th Conference on Uncertainty in Artificial Intelligence (UAI'98) , Madison, Wisconsin*, pages 1–7, San Francisco,California, juillet 1998. Morgan Kaufmann.

[2] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. In *Annals of Mathematics and Artificial Intelligence*, volume 34, 1-3, pages 197–215. Benferhat, Prade, eds Kluwer Academic Publishers, Dordrecht, The Netherlands, March 2002.

[3] Leila Amgoud and Laurent Perrussel. Arguments and Contextual Preferences. In *Computational Dialectics-Ecai workshop (CD2000) , Berlin*, August 2000.

[4] T.J.M. Bench-Capon. Value-based argumentation frameworks. In *Proc. of Non-monotonic Reasoning*, pages 444–453, 2002.

[5] Phan M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nomonotonic Reasoning and Logic Programming. In *Proc. of the 13th. International Joint Conference in Artificial Intelligence (IJCAI), Chambéry, Francia*, pages 852–857, 1993.

[6] Hadassa Jakobovits. Robust semantics for argumentation frameworks. *Journal of Logic and Computation*, 9(2):215–261, 1999.

[7] Diego C. Martínez, Alejandro J. García, and Guillermo R. Simari. On acceptability in abstract argumentation frameworks with an extended defeat relation. In *Proc. of I International Conference on Computational Models of Arguments, COMMA 2006*, pages 273–278, 2006.

[8] Diego C. Martínez, Alejandro J. García, and Guillermo R. Simari. Progressive defeat paths in abstract argumentation frameworks. In *Proceedings of the 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006*, pages 242–253, 2006.

[9] Diego C. Martínez, Alejandro J. García, and Guillermo R. Simari. Modelling well-structured argumentation lines. In *Proc. of International Joint Conference on Artificial Intelligence IJCAI-2007 (in press)*, 2007.

[10] M. Giacomin P. Baroni. Characterizing defeat graphs where argumentation semantics agree. In *ArgNMR, Workshop on Argumentation and Non-Monotonic Reasoning*, pages 33–48, 2007.

[11] Massimiliano Giacomin Pietro Baroni. Evaluation and comparison criteria for extension-based argumentation semantics. In *Computational Models of Argument - Proceedings of I International Conference on Computational Models of Arguments, COMMA 2006*, pages 157–168, 2006.

[12] Gerard A. W. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90(1–2):225–279, 1997.

Greedy Seeding Procedure for GAs Solving a Strip Packing Problem

Carolina Salto
Universidad Nacional de La Pampa
General Pico, Argentina
saltoc@ing.unlpam.edu.ar

and

Enrique Alba - Juan M. Molina
E.T.S.I. Informática, Universidad de Málaga
Málaga, España
{eat,jmmb}@lcc.uma.es

and

Guillermo Leguizamón
Universidad Nacional de San Luis
San Luis, Argentina
legui@unsl.edu.ar

Abstract

In this paper, the two-dimensional strip packing problem with 3-stage level patterns is tackled using genetic algorithms (GAs). We evaluate the usefulness of a greedy seeding procedure for creating the initial population, incorporating problem knowledge. This is motivated by the expectation that the seeding will speed up the GA by starting the search in promising regions of the search space. An analysis of the impact of the seeded initial population is offered, together with a complete study of the influence of these modifications on the genetic search. The results show that the use of an appropriate seeding of the initial population outperforms existing GA approaches on all the used problem instances, for all the metrics used, and in fact it represents the new state of the art for this problem.

Keywords: Genetic Algorithms, Strip Packing, Seeding

1 INTRODUCTION

The two-dimensional strip packing problem (2SPP) arise in many real-world applications such as in the paper or textil industries. Typically, the 2SPP consists of a set of M rectangular pieces, each defined by a *width* and a *height*, which have to be packed in a larger rectangle with a fixed width W and unlimited length, designated as the *strip*. The search is for a layout of all the pieces in the strip that minimizes the required strip length with the following restrictions: all pieces have to be packed with their sides parallel to the sides of the strip, without overlapping,

and rotations are not allowed. This problem is similar to the one of cutting the pieces out of the strip by means of orthogonal cuts, minimizing the consumed strip.

Additionally, another constraint is included in our problem: we consider only 3-stage level (guillotine) packing patterns. In these patterns, pieces are packed by horizontal levels (parallel to the bottom of the strip). Inside each level pieces are packed bottom left justified and, when there is enough room in the level, pieces with the same width are stacked one above the other. Many real application of 2D cutting and packing in the glass, wood, and paper industries consider 3-stage level patterns, for that the importance of incorporating this restriction in the problem formulation.

The 2SPP is NP-hard [9]. A few exact approaches for this problem are known [6, 14], although using metaheuristics is the usual way to solve it. Regarding the existing surveys of metaheuristics in the literature, Hopper and Turton [8, 9] review the approaches developed to solve 2D packing problems using GAs; Simulated Annealing, Tabu Search, and artificial Neural Networks are also considered. They conclude that Evolutionary Algorithms (EAs) [4, 15] are the most widely investigated meta-heuristics in the area of cutting and packing. Lodi et. al [13] consider several methods for the 2SPP in their survey and discussed also mathematical models; specially the case where the items have to be packed into rows forming levels are discussed in detail. Other representative works about GAs applied to solve level packing problems are [5], [10], [11], and [16] and 3-stage guillotine cuts [17, 18, 21, 22, 23]. The majority of the approaches imposing this last restriction deal with bin packing problems, for that the difficulty to find works in strip packing to compare with.

In this article we use a GA as the general driving force to locate the region in which a solution of minimum length is located. GAs deal with a population of tentative solutions, on which genetic operators are applied in an iterative manner to progressively compute new solutions of higher quality. We study a hybrid approach where a GA is combined with a heuristic placement routine. The GA is used to determine the sequence in which the pieces are to be packed, and the placement routine determines the layout of the pieces onto the strip in order to generate a 3-stage guillotine pattern.

The efficiency of a GA could be improved by increasing the quality of the initial population. This can be done by adopting simple rules. We here investigate the advantages of seeding the initial population using a set of greedy rules including information of the problem (such as the pieces width, the pieces area, etc.), resulting in a more specialized initial population. The main goal of this paper is to build an improved GA to solve larger problems than the ones found in the literature at present, and to quantify the effects of including a seeding procedure into the algorithms, looking for the best trade-off between exploration and exploitation in the search process, what is considered the key point to perform accurately and efficiently on complex applications.

The organization of the paper is as follows. The components of the used algorithm are described in Section 2. Section 3 describes the greedy generation of the initial population. In Section 4, we explain the parameter settings of the algorithms used in the experimentation. Section 5 reports on the algorithm performances. Finally, in Section 6, we give some conclusions and analyze future search directions.

2 A HYBRID GA FOR THE 2SPP

In Algorithm 1 we can see the structure of the basic steady-state GA ($(\mu + 1)$-GA) we use for solving the 2SPP. This algorithm creates an initial population $P(0)$ of μ solutions in a random

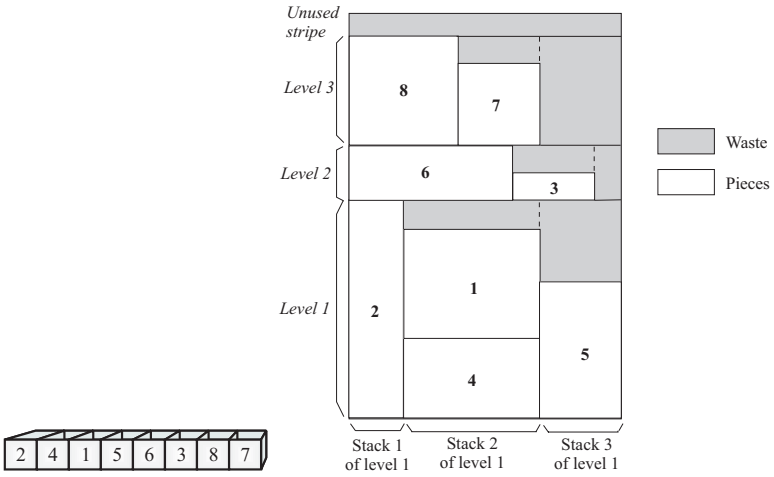


Figure 1: cutting pattern for the permutation 2 4 1 5 6 3 8 7.

(uniform) way, and then evaluates these solutions. The evaluation uses a placement algorithm to arrange the pieces in the strip to construct a feasible packing pattern. After that, the population goes into a cycle where it undertakes evolution. This cycle involves the selection of two parents by binary tournament and the application of recombination and mutation operators to create a new solution ($P'(t)$). The new generated individual replaces the worst individual in the population only if it is fitter. In this study, the stopping criterion for the cycle is to reach a maximum number of evaluations ($max\_evaluations$). The best solution is identified as the best individual ever found which minimizes the required strip length.

Algorithm 1 Genetic algorithm

```
GA
t = 0; {current generation}
initialize(P(t));
evaluate(P(t));
while (t < max_evaluations) do
   P'(t) = evolve(P(t)); {recombination and mutation}
   evaluate (P'(t));
   P(t + 1) = select(P'(t), P(t));
   t = t + 1;
end while
```

In most cases, applying a GA means devising a customized representation for encoding a candidate solution. We encode a packing pattern into a chromosome as a sequence of pieces that defines the input for the layout algorithm. Therefore, a chromosome will be a permutation $\pi = (\pi_1, \pi_2, ..., \pi_M)$ of M natural numbers (piece identifiers). In order to generate a 3-stage level pattern, a modified *next-fit* heuristic (NF) is used here —in the following referred as *modified next-fit*, or *MNF*— which was proved to be very efficient in [18, 20]. This heuristic gets a sequence of pieces as its input and constructs the packing pattern by placing pieces into stacks, and then stacks into levels in a greedy way, i.e., once a new stack or a new level is started, previous ones are never reconsidered. See Figure 1 for an illustrative example. Deeper explanation of the MNF procedure can be found in [23].

GAs are guided by the values computed by an objective function for each tentative solution until the optimum or an acceptable solution is found. In our problem, the objective is to minimize the strip length needed to build the layout corresponding to a given solution π. An important consideration is that two packing patterns could have the same length —so their fitness will be equal— although, from the point of view of reusing the trim loss, one of them

can be actually better because the trim loss in the last level (which still connects with the remainder of the strip) is greater than the one present in the last level of the other layout. Therefore we are using the following fitness function:

$$F(\pi) = strip.length - \frac{l.waste}{l.lenght * W} \quad (1)$$

where *strip.length* is the length of the packing pattern corresponding to the permutation π, $l.waste$ is the area of reusable trim loss in the last level l of the packing pattern and $l.lenght$ is the length of l. Hence, $F(\pi)$ is both simple and accurate.

Regarding the genetic operators, the recombination operator used here, *Best Inherited Level Recombination (BILX)* (introduced in [23] as BIL), incorporates some problem-specific knowledge into their mechanism in order to improve the trim loss. The BILX operator transmits the best levels of the parent to the child, i.e. those with the highest filling rate (fr) or, equivalently, with the least trim loss. This rate is calculated as follows, for a given level l:

$$fr(l) = \sum_{i=1}^{n} \frac{width(\pi_i) \times length(\pi_i)}{W \times l.length} \quad (2)$$

where $\pi_1, ..., \pi_n$ are the pieces in l, $width(\pi_i)$ and $length(\pi_i)$ are the piece dimensions.

Actually, BILX works as follows. Let be nl the number of levels in one parent $parent_1$. In the first step the filling rates of all nl levels from $parent_1$ are calculated. After that, a probability of selection, proportional to its filling rate, is assigned to each level and a number $(nl/2)$ of levels are selected from $parent_1$. The pieces π_i belonging to the inherited levels are placed in the first positions of the child. Meanwhile, the remaining positions are filled with the pieces which do not belong to that levels, in the order they appear in the other parent $parent_2$.

On the other side, the idea behind the mutation operator used in this work is to change the location of some pieces so that the final cost is reduced; it was successfully tested in [23]. Named as *Best and Worst Stripe Exchange (BW_SE)*, this mutation changes the location of the best and the worst level. The pieces of the best level (the one with highest filling rate) are allocated in the first positions of the new packing pattern while the pieces of the worst level are assigned to the last positions. The middle positions are filled with the remaining pieces in the order they appeared in the original packing pattern. In BW_SE, the movements can help to the involved levels or their neighbors to accommodate pieces from neighboring levels, thus improving their trim loss.

3 INITIAL SEEDING

The performance of a GA is often related to the quality of its initial population. This quality depends on two important issues: the average fitness of individuals in the population and the diversity in the population. By having an initial population with better fitness values, better final individuals can be found faster [1, 2, 19]. Besides, high diversity in the population inhibits early convergence to a locally optimal solution.

There are many ways to arrange this initial diversity. The idea in this work is to start with a seeded population created by following some building rules, hopefully allowing to reach good solutions in early stages of the search. The rules will include some characteristics from the problem such as piece sizes, and also incorporate ideas from the *best fit* (BF) and *first fit* (FF) heuristics [12].

Table 1: Rules to generate the initial population

#	Rule Description	#	Rule Description
1	sorts pieces by decreasing width.	2	sorts pieces by increasing width.
3	sorts pieces by decreasing length.	4	sorts pieces by increasing length.
5	sorts pieces by decreasing area.	6	sorts pieces by increasing area.

#	Rule Description
7	sorts pieces by alternating between decreasing width and length.
8	sorts pieces by alternating between decreasing width and increasing length.
9	sorts pieces by alternating between increasing width and length.
10	sorts pieces by alternating between increasing width and decreasing length.
11	the pieces are reorganized following the BF heuristic.
12	the pieces are reorganized following the FF heuristic.

Individuals are generated in two steps. In the first step, the packing patterns are randomly sampled from the search space with a uniform distribution. After that, each of them are modified by one rule, randomly selected, with the purpose of improving the piece location inside the random packing pattern. Each application of a rule yields a (possibly) different solution because of the randomization used in the first step.

The rules for the initial seeding are listed in the Table 1. These rules are proposed in order to produce individuals with improved fitness values and also for introducing diversity in the initial population. Hence, sorting the pieces by their width will hopefully increase the probability of stacking pieces, and then produce more dense levels. On the other hand, sorting by length will generate levels with smaller wasted space above the pieces, especially when the pieces lengths are very similar. BF and FF relocate the pieces with the goal of reducing the trim loss inside a level. The possible new layout obtained in this way has to be transmitted to the chromosome in such a way that we can obtain the same layout by applying MNF to the chromosome. Finally, rules 7 to 10 have been introduced for increasing the initial diversity. As we will see, these rules are not only useful for initial seeding: several of them can be used as a simple greedy algorithm for local search to help during the optimization process.

4 IMPLEMENTATION

The specific $(\mu+1)$-GA we have implemented is analyzed here using three different methods of seeding the initial population: (i) by means of a random generation (GA), (ii) by applying one determined rule from the Table 1 (GA_i where i stand for a rule number) and (iii) by applying a rule from Table 1, but a randomly selected one for each individual of the population (GA_{Rseed}). We also consider the random generation of individuals as other applicable rule in this case.

The population size was set to 512 individuals. The maximum number of evaluations was fixed to 2^{16}. The recombination operator was applied with a probability of 0.8, while the mutation probability was set to 0.1. Parameters (population size, stop criterion, probabilities, etc) were not chosen at random, but rather by an examination of values previously used with success (see [21]).

These algorithms were run in MALLBA [3], a C++ software library fostering rapid prototyping of hybrid and parallel algorithms, and the platform was an Intel Pentium 4 at 2.4 GHz and 512 MB RAM, linked by Fast Ethernet, under SuSE Linux with 2.4.19-4GB kernel version.

We have considered five randomly generated problem instances with M equal to 100, 150, 200, 250 and 300 pieces and a known global optimum equal to 200 (the minimal length of the strip). These instances belong to the subtype of the guillotine patterns only, i.e. the optimum value does not correspond to the 3-stage guillotine pattern constraint. They were obtained by

an own implementation of a data set generator, following the ideas proposed in [25] with the length-to-width ratio of all M rectangles in the range $1/3 \leq l/w \leq 3$. These instances are publicly available at http://mdk.ing.unlpam.edu.ar/ ˜lisi/2spp.htm.

5 COMPUTATIONAL ANALYSIS

In this section we summarize the results of applying the proposed algorithm with its seeding variants to all the problem instances. Our aim is to offer meaningful results and check them from a statistical point of view.

For each algorithm we have performed 30 independent runs per instance using the parameter values described in the previous section. Also, the evaluation considers two important issues for any search process: the capacity for generating new promising solutions and the pace rate of the progress in the surroundings of the best found solution (fine tuning or intensification). For a meaningful analysis we consider the average fitness values and the entropy measure (as proposed in [7]), which is computed as follows:

$$entropy = \frac{\sum_{i=1}^{M} \sum_{j=1}^{M} (\frac{n_{ij}}{\mu}) \ln(\frac{n_{ij}}{\mu})}{M \ln M} \quad (3)$$

where n_{ij} represents the number of times the piece i is set into the position j in the population of size μ. This function takes values in [0..1] and a value of 0 indicates that all the individuals in the population are identical.

Table 2 shows the results obtained for the different methods for generating the initial population for all instances (GA, GA_i and GA_{Rseed}). The most relevant metrics used in this comparison are: the average objective value of the initial population (column avg_{ini}), the best found feasible solution (column *best*) and the average value of the best found feasible solution, in the 30 independent runs, along with its standard deviation (column $avg_{\pm\sigma}$). The optimal *best* values are printed in bold.

From that results we can point out that any seeded GA starts the search process from better fitness values than in the case of randomly generated initial populations (GA). Best initial populations, in average, are obtained using GA_4 but having poor genetic diversity (see Figure 2 which shows the mean entropy values in the initial population for each algorithm and different M values). Rule 4 arranges the pieces by their height so the generated levels have very similar piece heights and consequently the produced free space inside a level is small. On the other side, a poor performance is obtained with both GA_7 and GA_9, which present in mean the worst initial population and also with poor genetic diversity. The random selection of rules to generate the initial solution (GA_{Rseed}) works quite well (being in the medium positions of the rule ranking) with an acceptable initial population diversity (entropy value near to 0.8).

Furthermore, a neat conclusion of this study is that the GA_{Rseed} (random seeded initial populations) outperforms significantly the GA with a traditional initialization in all the metrics (the p-value is close to 0) and also the rest of the seeded GAs. This suggests that the efficiency of a GA could be improved simply by increasing the quality of the initial population.

Figure 2 shows that the majority of the seeded GAs present poor initial genetic diversity (less than 0.5), except GA_{12}, GA_{11} and GA_{Rseed} with an initial diversity (near to 1) close to one of the no seeded GA. The two first algorithms apply NF and BF heuristics (respectively) to a random generated solution, i.e. the possibly changes in the piece positions generated by the heuristics does not take into account piece dimensions hence the original piece positions inside the chromosome are more probably maintained. On the other hand, GA_{Rseed} combine

Table 2: Experimental results for the GA regarding different seeding methods.

Alg	$M = 100$			$M = 150$			$M = 200$			$M = 250$			$M = 300$		
	avg_i	***best***	***avg***$\pm\sigma$	avg_i	***best***	***avg***$\pm\sigma$	avg_i	***best***	***avg***$\pm\sigma$	avg_i	***best***	***avg***$\pm\sigma$	avg_i	***best***	***avg***$\pm\sigma$
GA	417.34	235.75	248.13 ± 5.49	504.76	241.78	254.60 ± 6.29	485.28	249.64	258.50 ± 5.50	489.08	243.81	255.59 ± 7.35	531.22	253.75	263.77 ± 5.80
GA_1	353.24	240.68	253.25 ± 7.63	408.29	239.73	247.79 ± 4.76	382.10	243.70	250.72 ± 3.97	354.45	236.86	244.25 ± 6.06	409.46	285.50	295.50 ± 5.39
GA_2	298.24	231.67	241.05 ± 4.58	369.14	241.73	248.58 ± 4.52	352.55	232.83	243.65 ± 4.85	331.02	231.77	242.58 ± 4.78	345.53	251.68	261.55 ± 7.06
GA_3	326.18	264.20	266.20 ± 1.73	274.18	229.71	234.71 ± 3.19	295.05	230.64	233.64 ± 1.56	284.19	232.81	234.39 ± 1.08	273.28	245.71	245.96 ± 0.45
GA_4	282.59	230.20	231.12 ± 0.77	240.31	**228.71**	230.70 ± 0.81	244.43	231.58	231.96 ± 0.33	239.65	229.76	230.45 ± 0.35	239.13	234.59	234.70 ± 0.04
GA_5	381.31	342.41	346.56 ± 2.41	427.88	333.56	338.02 ± 2.26	415.83	295.63	329.08 ± 18.68	377.02	271.57	278.35 ± 4.28	401.03	326.50	329.00 ± 2.24
GA_6	371.87	294.29	299.42 ± 2.10	377.88	297.21	308.33 ± 4.74	359.04	265.59	291.18 ± 8.87	357.10	313.27	318.02 ± 2.42	375.12	321.53	324.97 ± 1.75
GA_7	397.19	256.58	265.98 ± 6.53	486.73	259.46	267.51 ± 4.33	452.62	252.74	272.19 ± 13.14	411.87	252.81	263.83 ± 8.06	454.54	281.71	287.63 ± 3.96
GA_8	298.95	240.28	242.03 ± 0.91	274.66	249.71	250.70 ± 0.60	275.21	240.62	241.50 ± 0.73	262.77	234.64	238.38 ± 2.19	259.60	246.43	246.49 ± 0.04
GA_9	384.78	255.63	261.95 ± 3.81	499.52	264.74	269.33 ± 4.36	460.18	248.54	261.45 ± 6.27	450.60	256.55	268.84 ± 4.74	486.68	280.60	294.07 ± 8.21
GA_{10}	354.03	262.47	279.10 ± 10.92	353.80	262.69	290.02 ± 12.62	368.99	258.55	271.16 ± 9.97	321.36	256.62	270.76 ± 10.97	336.15	275.73	287.45 ± 5.33
GA_{11}	277.87	229.78	236.66 ± 3.39	290.06	239.72	243.44 ± 2.05	278.44	226.80	234.69 ± 3.66	273.94	222.70	233.27 ± 4.08	283.25	229.58	237.57 ± 3.33
GA_{12}	281.23	233.77	240.05 ± 4.77	294.41	242.71	248.36 ± 4.49	282.96	231.69	237.84 ± 2.93	276.63	229.71	234.10 ± 3.22	287.98	227.67	241.60 ± 6.58
GA_{Rseed}	340.57	**228.59**	232.30 ± 1.86	369.50	229.04	231.66 ± 1.15	358.89	**220.71**	228.38 ± 3.98	340.64	**221.61**	227.36 ± 2.34	360.68	**225.49**	233.47 ± 2.60

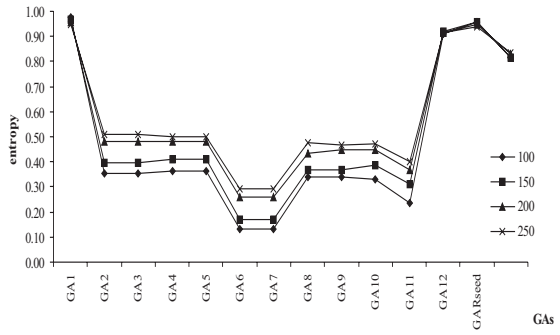


Figure 2: Average entropy of initial population.

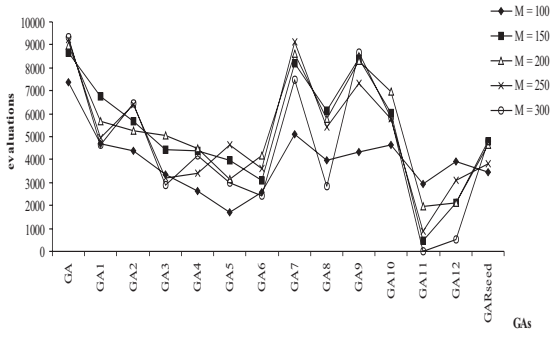


Figure 3: Mean number of evaluations to reach the best value for each instance.

all the previous considerations with random generation of the piece positions, so high genetic diversity was expected.

As supposed, the GA_{Rseed} significantly reduces the number of evaluations required to find good solutions (see Figure 3): they are near two time smaller in mean than GA. To confirm these observations, we used the t-test, which indicates that the difference among the algorithms are significant under this metric (p-values near to 0). GA_7, GA_9 and GA present similar effort in order to locate their best solutions: all of them have quite similar initial population averages (see Table 2) but GA presents best final solution qualities.

With respect to mean execution times there are a little difference against any seeded GA due to the initialization phase, but this difference is negligible.

After the 30 runs of GA_{Rseed} for all M values, we analyzed the contribution of the solutions

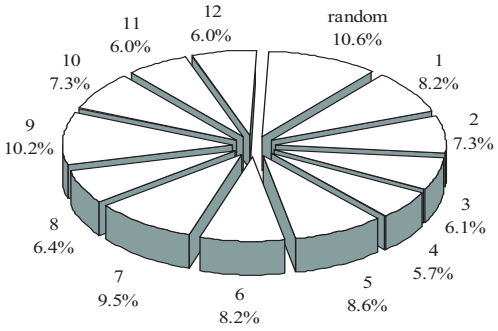


Figure 4: Rule Rank.

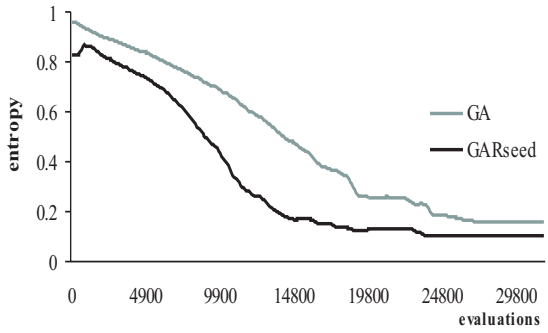


Figure 5: Population entropy for M=200.

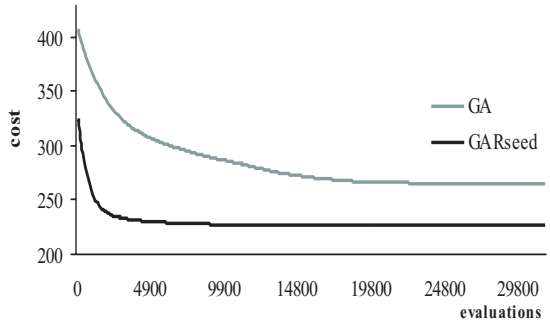


Figure 6: Average population fitness for M=200.

generated by each rule to the average fitness in the initial population. Regarding that measurement, the rule rank, from best to worst, was as follows: 13, 9, 7, 5, 1, 6, 10, 2, 8, 3, 12, 11 y 4 (see Figure 4), although the differences in percentages are quite small. The rule preserving the original piece position (random generation) are first in the rankings followed by the ones sorting pieces by alternating between increasing/decreasing width and height. By surprise, rules incorporating the BF and FF heuristics appear in last positions, when the expectation was that they could create good packing patterns due to the improvement in the layout they achieved. Also a remarkable point here is the rank position of rule 4 which appeared in the last position, although its use for the initialization of the whole population brings to a better performance than the rest.

5.1 Additional Discussion

If we seek among all the proposed algorithms, we can state that the GA_{Rseed}, which initializes the population considering the whole set of rules including information of the problem, converges to a better final solution, allowing a highly reduced number of evaluations comparing to that of GA using a random sampled initial population. So, for the next study we compare the performance of GA_{Rseed} with the performance of the plain GA.

These algorithms were compared with respect to the phenotypic entropy (Figure 5) as well as the evolution of the average population fitness (Figure 6). These comparisons are illustrated over the instance with M=200 (similar results are obtained with the rest of the instances). From these figures, we observe that the GA with seeding has a fast phenotypic entropy decrease to the

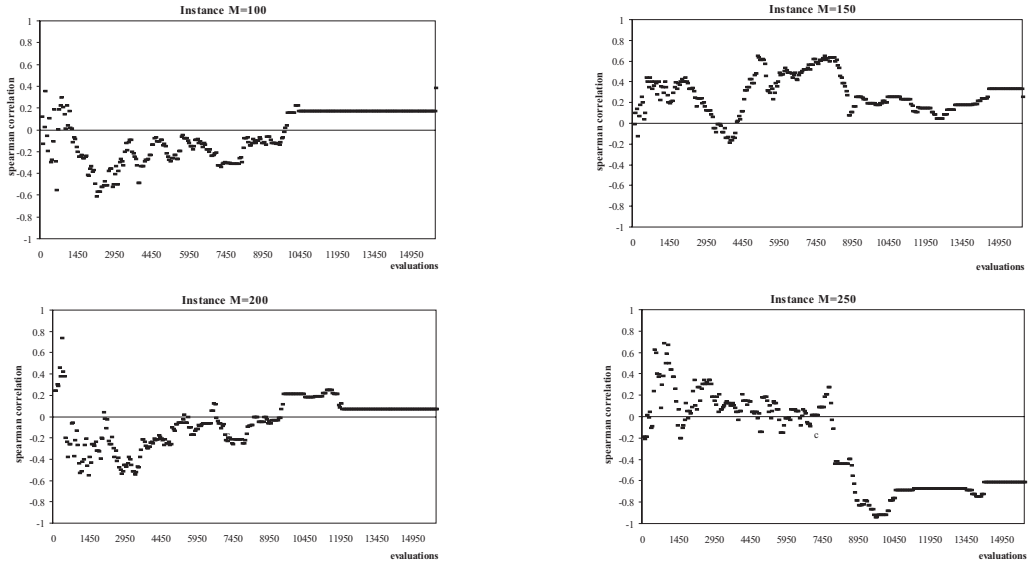


Figure 7: Evolving populations correlation between best fitness in each population and entropy measure. Each point represents the correlation between 15 populations from a 15 runs.

best final solutions (resulting in faster convergence), but the genetic diversity decreases slower than phenotypic diversity thus. In fact, we can see that the seeded initial populations present a high entropy value (near to 0.85 value), i.e., good diversity, but with fitter individuals.

In order to provide a deeper explanation of what is happening during the search of the algorithms we analyze the relationship between the fitness and the population entropy (genotypic diversity), which will have a strong effect on search difficulty. A way to examine this is by the use of the Spearman correlation measure (a nonparametric correlation test), which ranks two sets of variables and tests for a linear relationship between the variables rank. The Spearman correlation coefficient is computed (from [24]) as follows:

$$1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N^2 - N} \tag{4}$$

where N is the number of items, and d_i is the difference between each pair of fitness and entropy ranks. The correlation values range from +1 (perfect positive correlation), through 0 (no correlation), to -1.0 (negative correlation). For our study, if we see ideal low fitness values, which will be ranked in ascending order (1=best, ..., 15=worst) and high diversity, ranked in ascending order (1=lowest diversity and 15=highest diversity), then the correlation coefficient should be strongly negative. Alternatively a positive correlation indicates that either bad fitness accompanies high diversity or good fitness accompanies low diversity.

Figure 7 shows the correlation between entropy and best fitness for each generation. Note that each point represents the correlation between 15 populations, sampled from 15 runs where there is a dependency of later evaluations on preceding ones. Experiments with instance M=250 show a period of fluctuation until evaluation 8000, then the correlation coefficient became strongly negative, indicating that a good fitness is present with good diversity. Experiments with instances with M=100 and M=200 contain a period of fluctuation until evaluations 11000 approximately, after which entropy lost correlation with best fitness (coefficient correlations near to zero). For these two instances, the relationship between fitness and diversity becomes less important, probably due to other critical relationships. Finally, after a period of early

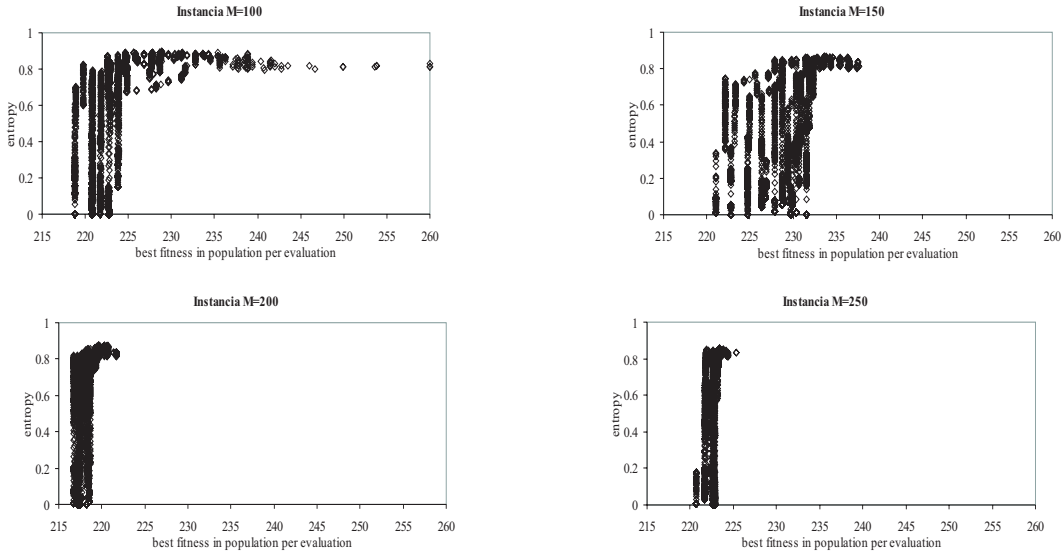


Figure 8: Best fitness plotted against that population's diversity.

fluctuations, the correlation for instance M=150 is positive, this can be owing to the fact that, as the best fitness is achieved early in a run, the population is made up of an increasingly larger number of copies of the best fit individual.

Figure 8 plots the best fitness found in the population along the x axis and population's diversity on the y axis. From there we can see some populations with bad fitness (low values are better) occurring with higher entropy. Once the algorithm find a good fitness, the entropy values begin to decrease (the populations are less diverse), i.e. the population is made up of an increasingly larger number of copies of the best fit individual.

6 CONCLUSIONS

In this paper we have investigated different greedy methods of generating the initial population in a traditional GA to solve the 3-stage 2SPP. The study, validated from a statistical point of view, analyzes the capacity of the seeding policy to generate new potentially promising individuals and the ability to keep a diversified population. The problem-aware seeding includes pieces dimensions and known level heuristics.

Our results show that an improvement in the GA performance was observed by using a problem-aware seeding, regarding both efficiency (effort) and quality of the solutions found. Moreover, the random selection of rules to build the initial population works properly, providing good genetic diversity of initial solutions. The performance of a GA is sensitive to the quality of its initial population. By having an initial population with better fitness values, we typically get better final individuals.

As future works we plan to go into the analysis of correlation between mean population fitness and genetic diversity in any depth. Also we propose to investigate non-permutation representations and a direct mapping to the final disposition of the pieces, as well as to construct parallel versions of the algorithms studied in this work.

ACKNOWLEDGEMENTS

This work has been partially funded by the Spanish Ministry of Education and the European FEDER under contract TIN2005-08818-C04-01 (the OPLINK project, http://oplink.lcc.uma.es). We also acknowledge the Universidad Nacional de La Pampa, and the ANPCYT in Argentina from which we received continuous support.

REFERENCES

[1] R.K. Ahuja and J.B. Orlin. Developing fitter genetic algorithms. *INFORMS Journal on Computing*, 9(3):251–253, 1997.

[2] R.K. Ahuja, J.B. Orlin, and A. Tiwari. A greedy genetic algorithm for the quadratic assignment problem. *Computers & Operations Research*, 27(3):917–934, 2000.

[3] E. Alba, J. Luna, L.M. Moreno, C. Pablos, J. Petit, A. Rojas, F. Xhafa, F. Almeida, M.J. Blesa, J. Cabeza, C. Cotta, M. Díaz, I. Dorta, J. Gabarró, and C. León. *MALLBA: A Library of Skeletons for Combinatorial Optimisation*, volume 2400 of *LNCS*, pages 927–932. Springer, 2002.

[4] T. Bäck, D. Fogel, and Z. Michalewicz. *Handbook of evolutionary computation*. Oxford University Press, New York, 1997.

[5] A. Bortfeldt. A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. *European Journal of Operational Research (article in press)*, 2005.

[6] S.P. Fekete and J. Schepers. On more-dimensional packing III: Exact algorithm. Technical Report ZPR97-290, Mathematisches Institut, Universität zu Köln, available from the first author at Department of Mathematics, 1997.

[7] J.J. Grefenstette. *Genetic Algorithms and Simulated Annealing*, chapter Incorporating problem specific knowledge into genetic algorithms, pages 42–60. Morgan Kaufmann Publishers, 1987.

[8] E. Hopper. *Two-dimensional packing utilising evolutionary algorithms and other meta-heuristic methods*. PhD thesis, University of Wales, Cardiff, U.K., 2000.

[9] E. Hopper and B. Turton. A review of the application of meta-heuristic algorithms to 2d strip packing problems. *Artificial Intelligence Review*, 16:257–300, 2001.

[10] S. Hwang, C. Kao, and J. Horng. On solving rectangle bin packing problems using genetic algorithms. *IEEE International Conference on Systems, Man, and Cybernetics - Humans, Information and Technology*, 2:1583–1590, 1994.

[11] B. Kroger. Guillotineable bin-packing: a genetic approach. *European Journal of Operational Research*, 84:645–661, 1995.

[12] A. Lodi, S. Martello, and M. Monaci. Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematicsournal of Operation Research*, 141:241–252, 2002.

[13] A. Lodi, S. Martello, and M. Monaci. Two-dimensional packing problems: a survey. *European Journal of Operational Research*, 141:241–252, 2002.

[14] S. Martello, S. Monaci, and D. Vigo. An exact approach to the strip-packing problem. *Informs Journal on Computing*, 15:310–319, 2003.

[15] M. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, third revised edition, 1996.

[16] C.L. Mumford-Valenzuela, J. Vick, and P.Y. Wang. *Metaheuristics: Computer Decision-Making*, chapter Heuristics for large strip packing problems with guillotine patterns: An empirical study, pages 501–522. Kluwer Academic Publishers BV, 2003.

[17] J. Puchinger and G.R. Raidl. Models and algorithms for three-stage two-dimensional bin packing. Technical report, Technische Universitat Wien, Institul fur Computergraphik und Algorithmen, 2004.

[18] J. Puchinger, G.R. Raidl, and G. Koller. *Solving a Real-World Glass Cutting Problem*, volume 3004 of *LNCS*, pages 162–173. Springer, 2004.

[19] C.R. Reeves. A genetic algorithm for flowshop sequencing. *Computers & Operations Research*, 22(1):5–13, 1995.

[20] C. Salto, J.M. Molina, and E. Alba. Sequential versus distributed evolutionary approaches for the two-dimensional guillotine cutting problem. *Proceedings of International Conference on Industrial Logistics (ICIL 2005)*, pages 291–300, 2005.

[21] C. Salto, J.M. Molina, and E. Alba. Analysis of distributed genetic algorithms for solving cutting problems. *International Transactions in Operational Research*, 13(5):403–423, 2006.

[22] C. Salto, J.M. Molina, and E. Alba. A comparison of different recombination operators for the 2-dimensional strip packing problem. *Proceedings of the XII Congreso Argentino de Ciencias de la Computacin (CACIC'06)*, 2006.

[23] C. Salto, J.M. Molina, and E. Alba. Evolutionary algorithms for the level strip packing problem. *Proceedings of the Workshop on Nature Inspired Cooperative Strategies for Optimization NICSO 2006*, pages 137–148, 2006.

[24] S. Siegel. *Nonparametric Statistics for the Behavioral Sciences*. New York: McGraw-Hill, 1956.

[25] P.Y. Wang and C.L. Valenzuela. Data set generation for rectangular placement problems. *EJOR*, 134:378–391, 2001.

Model Contractions on Description Logics*

Martín O. Moguillansky Marcelo A. Falappa

Consejo de Investigaciones Científicas y Técnicas (CONICET)
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de la Computación (DCIC)
Universidad Nacional del Sur (UNS)
Av. Alem 1253 - (B8000CPB) Bahía Blanca - Argentina
PHONE/FAX: (+54)(291)459-5136
E-MAIL: mom@cs.uns.edu.ar mfalappa@cs.uns.edu.ar

Abstract

When using *tableaux algorithms* to reason about *Description Logics* (DLs), new information is inferred from the models considered while trying to achieve knowledge satisfiability. By focusing the *ontology change* problem, we consider an environment where DLs are the logical formalization to express knowledge bases in the web, and the integration of distributed ontologies is developed under new extensions of the *belief revision* theories originally exposed in [1]. Hence, a reinforced theory arises in order to properly apply change operations over models, considering new inferred information and assumed beliefs in each *possible world*. As a result, a new type of *contraction* operator is proposed and its success postulate analyzed.

Keywords: Belief Revision, Description Logics, Tableau Calculi, Ontology Change.

1 Introduction

Our main research interest relays in topics like *Ontology Integration* and *Ontology Merging* [4], for what we have proposed in [7] to use theory change formalizations in order to consistently join terminologies, redefining or reinforcing sub-concepts. But following the reasoning methods exposed for DLs, like satisfiability, solved by tableaux algorithms originally defined in [8], a new area of interest arises. A set of knowledge base extensions is obtained from the models considered during the execution of the DL reasoning service. Here, is imperative to redefine the formalizations of the theory change in order to revise beliefs on each extension and transitively in the knowledge base itself.

A motivating environment in which our proposal seems relevant may be the case of large databases (closed world assumption) managing incomplete information. This means that for some systems, maybe some unnoticed information is inferred from the knowledge base. Moreover, it is possible to have critical information deduced from the base that may take over more undesirable deductions. In order to avoid this kind of scenarios, a database manager could classify those basic beliefs that should not be inferred from the base, and present them as complex queries to the base. In case that some query is verified, he could have the alternative to correct the knowledge that helps to get this

*This article assumes some background knowledge about description logics and belief revision from the reader.

Partially financed by CONICET (PIP 5050), Universidad Nacional del Sur (PGI 24/ZN11) and Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096).

deduction, *i.e.*, to contract some beliefs in order to avoid problematic deductions. Moreover, this could motivate the writing of some kind of basic rules, possibly exposed as axioms representing some basic knowledge to always be held no matter the upgrade needed to be done to the base.

This is the situation in which we focus our work, anticipating or foretelling undesirable side-effects and / or clashes in large database systems. While some slight details could pass unnoticed by the ordinary people's inspection, the proposed theory could inspect possible worlds by looking some specific assumed beliefs in order to infer new knowledge from the basic knowledge representation. The remainder of this work is disposed as follows, Sect. 2 gives a brief description of tableau-based algorithms and their behavior by achieving satisfiability. Sect. 3 gives a brief overview of kernel contractions in the theory change introducing Sect. 4, which explains the proposed theory where some of the basic definitions for kernel contractions were adapted to deal with model based reasoners. Finally, Sect. 5 concludes making an analysis of the proposal.

2 DLs Reasoning Algorithms

Relevant inference problems usually are reduced to the consistency problem for ABoxes, provided that the DL at hand allows for conjunction and negation. However, for those description languages of DL systems that do not allow for negation, subsumption of concepts can be computed by so-called *structural subsumption algorithms*, *i.e.*, algorithms that compare the syntactic structure of (possibly normalized) concept descriptions.

While usually very efficient, they are only complete for rather simple languages with little expressivity. In particular, DLs with (full) negation and disjunction cannot be handled by structural subsumption algorithms. For such languages, so-called *tableau-based algorithms* have turned out to be very useful.

2.1 Properties for Reasoning

Let first give a very brief description of some important reasoning properties of description logics. Given a terminology $\mathcal{T}$, if there is some interpretation of a concept that satisfies the axioms in $\mathcal{T}$ (a model of $\mathcal{T}$), then the concept denotes a nonempty set for the interpretation, furthermore this concept is known to be *satisfiable* w.r.t. $\mathcal{T}$. Otherwise it is called *unsatisfiable*. Formally,

(Satisfiability) [2] A concept C is *satisfiable* w.r.t. $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}}$ is nonempty. In such a case we say that $\mathcal{I}$ is a *model* of C.

Checking (un)satisfiability of concepts might be considered a key inference given that a number of other important inferences for concepts can be reduced to it. For instance, in order to check whether a domain model is correct, or to optimize concepts, we may want to know whether one concept is more general than another. This is called the *subsumption* problem. A concept C is *subsumed* by a concept D if in every model of $\mathcal{T}$, C is a subset of D.

(Subsumption) [2] A concept C is *subsumed* by a concept D w.r.t. $\mathcal{T}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$. In such a case we write $C \sqsubseteq_{\mathcal{T}} D$ or $\mathcal{T} \models C \sqsubseteq D$.

A new kind of reasoning algorithms in DLs raised from the approach of considering satisfiability checking as the main inference. These algorithms are known as tableaux and can be understood as a specialized tableau calculi.

2.2 Basics for Tableaux Algorithms

Instead of directly testing subsumption of concept descriptions, these algorithms use negation to reduce subsumption to (un)satisfiability of concept descriptions: $C \sqsubseteq D$ ***iff*** $C \sqcap \neg D$ ***is unsatisfiable.***

We illustrate the underlying ideas by two simple examples taken from [2]. Let A, B be concept names, and let R be a role name. As a first example, assume that we want to know whether $(\exists R.A) \sqcap (\exists R.B)$ is subsumed by $\exists R.(A \sqcap B)$. This means that we must check whether the concept description $C = (\exists R.A) \sqcap (\exists R.B) \sqcap \neg(\exists R.(A \sqcap B))$ is unsatisfiable.

Pushing all negation signs as far as possible into the description yields $C_0 = (\exists R.A) \sqcap (\exists R.B) \sqcap \forall R.(\neg A \sqcup \neg B)$, which is in negation normal form, *i.e.*, negation occurs only in front of concept names. Then, we try to construct a finite interpretation $\mathcal{I}$ such that $C_0^{\mathcal{I}} \neq \emptyset$. This means that there must exist an individual in $\Delta^{\mathcal{I}}$ that is an element of $C_0^{\mathcal{I}}$. The algorithm just generates such an individual, say b, and imposes the constraint $b \in C_0^{\mathcal{I}}$ on it, this means that b must satisfy all the three interpreted conjunctions that composes C_0.

From $b \in (\exists R.A)^{\mathcal{I}}$ we can deduce that there must exist an individual c such that $(b, c) \in R^{\mathcal{I}}$ and $c \in A^{\mathcal{I}}$. Analogously, $b \in (\exists R.B)^{\mathcal{I}}$ implies the existence of an individual d with $(b, d) \in R^{\mathcal{I}}$ and $d \in B^{\mathcal{I}}$. In this situation, one should not assume that $c = d$. Thus:

- *For any existential restriction the algorithm introduces a new individual as role filler, and this individual must satisfy the constraints expressed by the restriction.*

Since b must also satisfy the value restriction $\forall R.(\neg A \sqcup \neg B)$, and c, d were introduced as R-fillers of b, we obtain the additional constraints $c \in (\neg A \sqcup \neg B)^{\mathcal{I}}$ and $d \in (\neg A \sqcup \neg B)^{\mathcal{I}}$. Thus:

- *The algorithm uses value restrictions in interaction with already defined role relationships to impose new constraints on individuals.*

Now c might be such that $c \in (\neg A)^{\mathcal{I}}$ or $c \in (\neg B)^{\mathcal{I}}$. Assume the first possibility leads to an obvious contradiction, so we must choose the second one $c \in (\neg B)^{\mathcal{I}}$. Analogously, we must choose $d \in (\neg A)^{\mathcal{I}}$ in order to satisfy the constraint $d \in (\neg A \sqcup \neg B)^{\mathcal{I}}$ without creating a contradiction to $d \in B^{\mathcal{I}}$. Thus:

- *For disjunctive constraints, the algorithm tries both possibilities in successive attempts. It must backtrack if it reaches an obvious contradiction, i.e., if the same individual must satisfy constraints that are obviously conflicting.*

In the example, we have now satisfied all the constraints without encountering an obvious contradiction. This shows that C_0 is satisfiable, and thus $(\exists R.A) \sqcap (\exists R.B)$ is not subsumed by $\exists R.(A \sqcap B)$. The interpretation generated by the algorithm is $\Delta^{\mathcal{I}} = \{b, c, d\};\ R^{\mathcal{I}} = \{(b, c), (b, d)\};\ A^{\mathcal{I}} = \{c\}$ and $B^{\mathcal{I}} = \{d\}$.

In our second example, we now want to know whether $(\exists R.A) \sqcap (\exists R.B) \sqcap \leqslant 1R$ is subsumed by $\exists R.(A \sqcap B)$. The tableau-based satisfiability algorithm first proceeds as above, with the only difference that there is the additional constraint $b \in (\leqslant 1R)^{\mathcal{I}}$. In order to satisfy this constraint, the two R-fillers c, d of b must be identified with each other. Thus:

- *If an at-most number restriction is violated then the algorithm must identify different role fillers.*

The individual $c = d$ must belong to both $A^{\mathcal{I}}$ and $B^{\mathcal{I}}$, which together with $c = d \in (\neg A \sqcup \neg B)^{\mathcal{I}}$ always leads to a clash. Thus, the search for a counterexample to the subsumption relationship fails, and the algorithm concludes that $(\exists R.A) \sqcap (\exists R.B) \sqcap \leqslant 1R \sqsubseteq \exists R.(A \sqcap B)$.

3 Kernel Contractions

A *belief base* is a knowledge state represented by a set of sentences not necessarily closed under logical consequence. Similarly, a *belief set* is a set of sentences closed under logical consequence. In general, a belief set is infinite, being this the main reason of the impossibility to deal with this kind of sets in a computer. Instead, it is possible to characterize the properties that must satisfy each of the change operations on finite representations of a knowledge state.

The classic operations in the theory change [1] are *expansions*, *contractions*, and *revisions*. An *expansion*, noted with "$+$", adds a new belief to the epistemic state, without guaranteeing its consistency after the operation. A *contraction*, noted with "$-$", eliminates a belief α from the epistemic state and, some of those beliefs that make possible its deduction. The sentences to eliminate might represent the *minimal change* on the epistemic state. Finally, a *revision* , noted with "$*$", inserts sentences to the epistemic state, guaranteeing consistency (if it was consistent before the operation). This means that a revision adds a new belief and perhaps it eliminates others in order to avoid inconsistencies.

The *kernel contraction* operator is identically applicable to belief bases and sets. It consist of a contraction operator capable of the selection and elimination of those beliefs in K that contribute to infer α.

Definition 3.1 - Set of Kernels [6]: Let K be a set of sentences and α a sentence. The set $K^{\perp\!\!\!\perp}\alpha$, called *set of kernels* is the set of sets K' such that (1) $K' \subseteq K$, (2) $K' \vdash \alpha$, and (3) if $K'' \subset K'$ then $K'' \nvdash \alpha$. The set $K^{\perp\!\!\!\perp}\alpha$ is also called *set of* α*-kernels* and each one of its elements are called α*-kernel*.

For the success of a contraction operation, we need to eliminate, at least one element of each α-kernel. The elements to be eliminated are selected by an *incision function*.

Definition 3.2 - Incision Function [6]: Let K be a set of sentences and "σ" be an *incision function* for it such that for any sentence α it verifies, (1) $\sigma(K^{\perp\!\!\!\perp}\alpha) \subseteq \bigcup(K^{\perp\!\!\!\perp}\alpha)$ and (2) If $K' \in K^{\perp\!\!\!\perp}\alpha$ and $K' \neq \emptyset$ then $K' \cap \sigma(K^{\perp\!\!\!\perp}\alpha) \neq \emptyset$.

Definition 3.3 - Kernel Contraction Determined by "σ" **[6]:** Let K be a set of sentences, α a sentence, and $K^{\perp\!\!\!\perp}\alpha$ the set of α-kernels of K. Let "σ" be an incision function for K. The operator " $-_\sigma$ ", called *kernel contraction determined by* "σ", is defined as, $K -_\sigma \alpha = K \backslash \sigma(K^{\perp\!\!\!\perp}\alpha)$.

Finally, an operator " $-$ " is a kernel contraction operator for K if and only if there exists an incision function "σ" such that $K - \alpha = K -_\sigma \alpha$ for all sentence α.

4 Model Contractions

In formalisms like description logics, the reasoning service is model based, *i.e.*, in order to make deductions the reasoning service looks into every possible world, this means that new deductions are made not only by a classical inference operator "$\vdash$", but also by applying it over every model of the base. Therefore, and considering the motivation exposed in the introductory section of this paper, our interest relays in checking satisfiability for a given possibly complex query α, and considering an affirmative response, breaking thereafter its trueness by applying a base contraction by α .

In this sense, let first analyze the scope of a modeling inference operator "$\models$" such that,

$$\textbf{(Entailment)}\ \Sigma \models \alpha \textit{ iff } \mathcal{M}(\Sigma) \subseteq \mathcal{M}(\{\alpha\})$$

where $\mathcal{M}(\Sigma)$ makes reference to the set of models of a knowledge base Σ, and $\mathcal{M}(\{\alpha\})$ identifies the set of models for a valid sentence α of the language.

Afterwards, a query α to the base Σ, noted as $\Sigma \models^{?} \alpha$, is solved by using a satisfiability checking process. This is done by generating all possible interpretations that satisfies every sentence in the base Σ, *i.e.*, finding every model for the base Σ. Afterwards, the reasoning process checks for every element in $\mathcal{M}(\Sigma)$, *i.e.*, if every model of the base, let say $\mathcal{M}_i$ is also a model of α. This is exactly the entailment definition.r If this is true, the query is said to be verified and $\Sigma \models \alpha$, *i.e.*, that the base infers α, being `YES` the answer for the query. If the base verifies $\neg\alpha$ then it is said that the query is not verified, being `NO` its answer. The third possibility would be known as an indecision and answered with an `UNKNOWN` if it is not verified α nor $\neg\alpha$.

As seen in Sect. 2.2, new knowledge may be inferred from assumptions made as a consequence of the applied satisfiability process following a tableau-based algorithm. From now on we focuss our attention in checking and adapting the theory change definitions cited in Sect. 3 applied to these extensions.

4.1 Extended Set of Kernels

Following the definitions cited in Sect. 3, a proof for a belief α is given by a set of minimal proofs or α-kernels. But this proofs are enclosed inside the knowledge base Σ itself. By using model based reasoning services, beliefs outside the scope of the base are assumed, exceeding the basic *set of kernels* given in Def. 3.1. Thereafter some definitions of the belief revision model should be redefined in order to be adapted to this new theory. In this sense, let first refer to each model $\mathcal{M}_i$ as an extension of the base Σ from now on noted as Σ_i identified as its i^{th} extension as follows.

Definition 4.1.1 - Base Extension: Let Σ be a knowledge base and $\mathcal{M}(\Sigma) = \{\mathcal{M}_1, ..., \mathcal{M}_n\}$ the set of $\mathcal{M}_i$ models of Σ, where $1 \leq i \leq n$, *i.e.*, $|\mathcal{M}(\Sigma)| = n$. The i^{th} base extension with respect to the model $\mathcal{M}_i$ is identified as Σ_i.

A model $\mathcal{M}_i$ may include several minimal proofs K for a belief α, *i.e.*, $K \subseteq \mathcal{M}_i$, where $\mathcal{M}_i \in \mathcal{M}(\Sigma)$. This also means that every α-proof K is part of a base extension Σ_i and indeed, K is also an element of the set of proofs of that base extension, *i.e.*, $K \in \Sigma_i{}^{\Uparrow}\alpha$. Moreover, as part of an α-proof K there are *effective* beliefs belonging to the base Σ (this is a subset $K_\Sigma \subseteq K$) and some other *assumed* beliefs that are not part of the base Σ but of some model $\mathcal{M}_i$ of it (equivalently a subset $K_{\mathcal{M}_i} \subseteq K$). Therefore, an α-proof K is such that $K = K_\Sigma \cup K_{\mathcal{M}_i}$. This motivates the following definition by reinforcing the original set of kernels and their components.

Definition 4.1.2 - Extended Set of Kernels: Let Σ be a knowledge base and α be a sentence. The condition $\Sigma \models \alpha$ holds *iff* there exists a set $\Sigma^{\Uparrow}\alpha$ (namely *extended set of kernels*) of *non-empty* sets $\Sigma_i{}^{\Uparrow}\alpha$ (called i^{th} *set of extended* α*-kernels*) where each $\Sigma_i{}^{\Uparrow}\alpha$ is a set of sets K (namely *extended* α*-kernel*) such that the following conditions hold:

(1) $K = K_\Sigma \cup K_{\mathcal{M}_i}$, such that
   $i.$ $K_\Sigma \subseteq \Sigma$
   $ii.$ $K_{\mathcal{M}_i} \subseteq \mathcal{M}_i$ and $K_{\mathcal{M}_i} \not\subseteq \Sigma$

(2) $K \vdash \alpha$

(3) If $K' \subset K$ then $K' \nvdash \alpha$

The following observation relates an extended set of kernels $\Sigma^{\Uparrow}\alpha$ and the set of models $\mathcal{M}(\Sigma)$ for a knowledge base Σ in terms of cardinality.

Observation 4.1.3: If $\Sigma^{\perp\!\!\!\perp}\alpha$ is an *extended set of kernels* then $|\Sigma^{\perp\!\!\!\perp}\alpha| = |\mathcal{M}(\Sigma)|$.
Proof:

Supposing by the contrary that $|\Sigma^{\perp\!\!\!\perp}\alpha| = |\mathcal{M}(\Sigma)|$ does not hold, then two possibilities arise:

(1) If $|\Sigma^{\perp\!\!\!\perp}\alpha| < |\mathcal{M}(\Sigma)|$ then there exists a model $\mathcal{M}_i \in \mathcal{M}(\Sigma)$ such that there is no $K \subseteq \mathcal{M}_i$ where $K \vdash \alpha$. Therefore, the related i^{th} *set of extended* α*-kernels* is such that ${\Sigma_i}^{\perp\!\!\!\perp}\alpha = \emptyset$, but as we know by Def. 4.1.2 this is not plausible, thereafter ${\Sigma_i}^{\perp\!\!\!\perp}\alpha \notin \Sigma^{\perp\!\!\!\perp}\alpha$. Besides, by Def. 4.1.2 a condition $\mathcal{M}(\Sigma) \subseteq \mathcal{M}(\{\alpha\})$ holds, therefore $\mathcal{M}_i \in \mathcal{M}(\{\alpha\})$, then necessarily there is a $K \subseteq \mathcal{M}_i$ where $K \vdash \alpha$. *(ABSURD).*

(2) If $|\Sigma^{\perp\!\!\!\perp}\alpha| > |\mathcal{M}(\Sigma)|$ then there exists a i^{th} *set of extended* α*-kernels* such that ${\Sigma_i}^{\perp\!\!\!\perp}\alpha \in \Sigma^{\perp\!\!\!\perp}\alpha$, with $K \in {\Sigma_i}^{\perp\!\!\!\perp}\alpha$, where $K \vdash \alpha$. This is verified in a model $\mathcal{M}_i$ in which ${\Sigma_i}^{\perp\!\!\!\perp}\alpha$ is applied, such that $\mathcal{M}_i \notin \mathcal{M}(\Sigma)$. This implies that $\mathcal{M}_i$ is not a model for Σ, but $\mathcal{M}_i$ is a model for α due to the existence of $K \subseteq \mathcal{M}_i$ where $K \vdash \alpha$, therefore $\mathcal{M}(\Sigma) \not\subseteq \mathcal{M}(\{\alpha\})$. *(ABSURD).* ■

An α-proof K in a model based system may or may not contain assumed beliefs. Note that a knowledge base Σ deducing α, *i.e.*, $\Sigma \models \alpha$ may, in fact, also verify $\Sigma \vdash \alpha$. This means that there is at least one proof that does not need to use any assumed beliefs to achieve its validity. The following definition is given in order to clearly identify the two different kinds of α-proofs in a model based system.

Definition 4.1.4 - Possible (Effective) α**-Proof:** Let $\Sigma^{\perp\!\!\!\perp}\alpha$ be an extended set of kernels as specified in Def. 4.1.2 for a given sentence α, and ${\Sigma_i}^{\perp\!\!\!\perp}\alpha$ the i^{th} set of extended α-kernels K in it contained such that $K = K_\Sigma \cup K_{\mathcal{M}_i}$. Then an α-proof K is referred as a *Possible (*resp. of *Effective)* α*-Proof iff* $K_{\mathcal{M}_i} \subseteq K$ is such that $K_{\mathcal{M}_i} \neq \emptyset$ (resp. of $K_{\mathcal{M}_i} = \emptyset$).

Observation 4.1.5: If K is an effective α-proof in Σ then $\Sigma \vdash \alpha$
Proof:

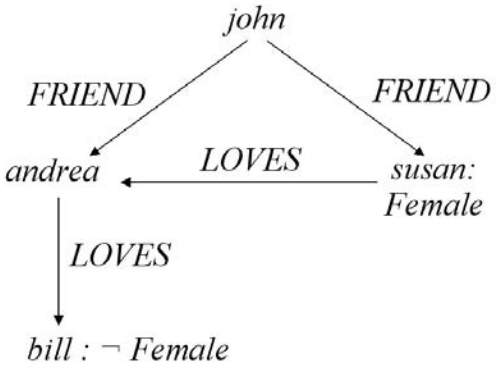
If K is an effective α-proof then by Def. 4.1.4 it follows that $K_{\mathcal{M}_i} = \emptyset$ for $K_{\mathcal{M}_i} \subseteq K$, hence $K \vdash \alpha$, and since $K \subseteq \Sigma$ and the consequence operator verifies *monotony*[1], it follows $\Sigma \vdash \alpha$. ■

Remark 4.1.6: All considered examples in this paper are reduced in a sense of relevance considering the extended α-proofs. This means that, although it is true that $|\Sigma^{\perp\!\!\!\perp}\alpha| = |\mathcal{M}(\Sigma)|$, since it is possible to have ${\Sigma_i}^{\perp\!\!\!\perp}\alpha, {\Sigma_j}^{\perp\!\!\!\perp}\alpha \in \Sigma^{\perp\!\!\!\perp}\alpha$ such that ${\Sigma_i}^{\perp\!\!\!\perp}\alpha \subseteq {\Sigma_j}^{\perp\!\!\!\perp}\alpha$, those ${\Sigma_i}^{\perp\!\!\!\perp}\alpha$ self contained by other i^{th} *set of extended* α*-kernel* are considered not relevant, and finally discarded.

The following example, borrowed from [5], shows the behavior of the new defined theory.

Example 1 : Given a knowledge base Σ as follows,

$$\Sigma = \left\{ \begin{array}{l} FRIEND(john, susan) \\ FRIEND(john, andrea) \\ LOVES(susan, andrea) \\ LOVES(andrea, bill) \\ Female(susan) \\ \neg Female(bill) \end{array} \right\}$$



[1] **(Monotony)** If $\Sigma' \subseteq \Sigma$ then $C_n(\Sigma') \subseteq C_n(\Sigma)$, for a given operator C_n such that $\alpha \in C_n(\Sigma)$ *iff* $\Sigma \vdash \alpha$.

we want to know if $john$ have a $Female\ FRIEND$ who $LOVES$ a male (*i.e.*, not $Female$) person. This is a query $\Sigma \models^{?} \alpha$ such that α is $\exists FRIEND.(Female \sqcap (\exists LOVES.\neg Female))(john)$. Following the given tableau specifications as a model based reasoning service we have two different models in order to achieve satisfiability of α, furthermore, the extended set of kernels would be such that $\Sigma^{\perp\!\!\!\perp}\alpha = \{{\Sigma_1}^{\perp\!\!\!\perp}\alpha, {\Sigma_2}^{\perp\!\!\!\perp}\alpha\}$, where each ${\Sigma_i}^{\perp\!\!\!\perp}\alpha$ is directly related to each model $\mathcal{M}_i$. Note that in this example there is only one α-proof K in each extended set of α-kernels.

$\mathcal{M}_1)$ $Female^{\mathcal{I}} = \{\}, \neg Female^{\mathcal{I}} = \{andrea\}$
${\Sigma_1}^{\perp\!\!\!\perp}\alpha = \{\{FRIEND(john, susan), Female(susan), LOVES(susan, andrea), \neg Female(andrea)\}\}$
$K_{\mathcal{M}_1} = \{\neg Female(andrea)\}$

$\mathcal{M}_2)$ $Female^{\mathcal{I}} = \{andrea\}, \neg Female^{\mathcal{I}} = \{\}$
${\Sigma_2}^{\perp\!\!\!\perp}\alpha = \{\{FRIEND(john, andrea), Female(andrea), LOVES(andrea, bill), \neg Female(bill)\}\}$
$K_{\mathcal{M}_2} = \{Female(andrea)\}$

□

4.2 Model Selection & Model Incision Function

Let think about an incision function cutting beliefs from a set of kernels in order to achieve a contraction of Σ by α, *i.e.*, we want to get some applied contraction operator " $\ominus$ " based on an incision function "σ" such that the query α turns to fail in the resultant knowledge base, this means that $\Sigma \ominus_{\sigma} \alpha \not\models \alpha$.

In order to achieve such an operation we only need to select one model, *i.e.*, an extension Σ_i, from which to break every α-proof in it such that the erasure is to be done from every K in ${\Sigma_i}^{\perp\!\!\!\perp}\alpha$ for some i. Moreover, those deleted beliefs from an α-proof should be "*effective*" beliefs, *i.e.*, beliefs from K_Σ, not assumed ones. The intuition in this is that no assumed belief could be cut off the knowledge base just because it is not part of the base. No justification can be supported in order to modify the knowledge base by making a specific *possible world* the new epistemic state.

In this sense, let first define a *model selection function* ρ, that following some *preference criterion* among the considered models, it takes the "*most conservative selection*" such that among every considered model, the selected one is the best choice to make a suitable further incision. Formally,

Definition 4.2.1 - Model Selection Function: Let Σ be a knowledge base and $\Sigma^{\perp\!\!\!\perp}\alpha$ an *extended set of kernels* for a valid sentence α, then a function "ρ" is a *model selection function* determined by some *preference criterion* such that $\rho(\Sigma^{\perp\!\!\!\perp}\alpha) = {\Sigma_i}^{\perp\!\!\!\perp}\alpha$, where ${\Sigma_i}^{\perp\!\!\!\perp}\alpha$ is valid in a model $\mathcal{M}_i$ considered the "*less relevant model*" of Σ.

Inspired in Def. 3.2, we propose a "*model incision function*" determined by a *model selection function* as follows,

Definition 4.2.2 - Model Incision Function: Let Σ be a knowledge base, $\Sigma^{\perp\!\!\!\perp}\alpha$ an *extended set of kernels* for a valid sentence α, and ρ a *model selection function*, then a function "σ" is defined as a *model incision function* such that it verifies,

(1) $\sigma(\rho(\Sigma^{\perp\!\!\!\perp}\alpha)) \subseteq \bigcup(\rho(\Sigma^{\perp\!\!\!\perp}\alpha)) \cap \Sigma$

(2) If $K_\Sigma \subseteq K \in \rho(\Sigma^{\perp\!\!\!\perp}\alpha)$ then $K_\Sigma \cap \sigma(\rho(\Sigma^{\perp\!\!\!\perp}\alpha)) \neq \emptyset$

Example 2 : Following the proposed definition for the *model incision function*, let continue with Ex. 1. As explained before, an incision function cuts beliefs only from one base extension in order

to break all proofs in the possible world by it determined. In this sense, suppose a *model selection function* selects the first base extension as the *most suitable model*, such that $\rho(\Sigma^{\perp\!\!\!\perp}\alpha) = {\Sigma_1}^{\perp\!\!\!\perp}\alpha$. Therefore, a *model incision function* would be applied to it in order to cut beliefs from every proof in the model $\mathcal{M}_1$. Finally a possibility would be $\sigma(\rho(\Sigma^{\perp\!\!\!\perp}\alpha)) = \sigma({\Sigma_1}^{\perp\!\!\!\perp}\alpha) = \{Female(susan)\}$. [2]□

4.3 Model Contraction Operator

In what follows let define the "*model contraction operator*" by reinforcing the kernel contractions as exposed in Def. 3.3, then a model contraction operator " $\ominus$ " is determined by a *model incision function* "σ" as follows,

Definition 4.3.1 - Model Contraction Determined by σ: Let Σ be a knowledge base, α be a sentence, "σ" be a *model incision function* determined by a *model selection function* ρ for Σ, and $\Sigma^{\perp\!\!\!\perp}\alpha$ be the extended set of α-kernels of Σ to which the function ρ is applied. The operator " $\ominus_\sigma$", referred as *model contraction determined by* "σ", is defined as,

$$\Sigma \ominus_\sigma \alpha = \Sigma \backslash \sigma(\rho(\Sigma^{\perp\!\!\!\perp}\alpha))$$

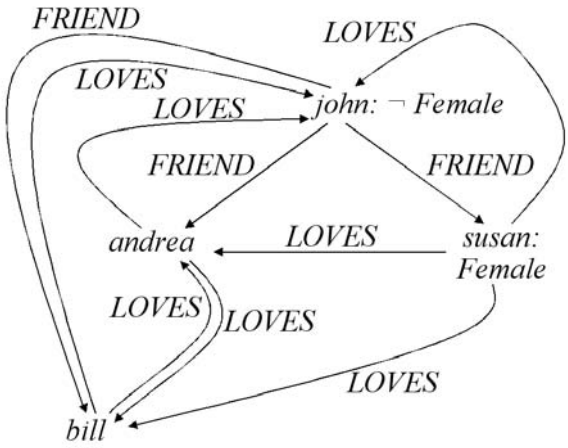
Finally, an operator " $\ominus$ " is a *model contraction* operator for Σ if and only if there exists a *model incision function* "σ" such that $\Sigma \ominus \alpha = \Sigma \ominus_\sigma \alpha$ for all sentence α.

Example 3 : Let conclude with the Ex. 1, and finally apply the contraction operator $\ominus$ to the selection made before in Ex. 2, where $\sigma(\rho(\Sigma^{\perp\!\!\!\perp}\alpha)) = \{\neg Female(susan)\}$. Then from Def. 4.3.1 follows that, $\Sigma \ominus_\sigma \alpha = \Sigma \backslash \sigma(\rho(\Sigma^{\perp\!\!\!\perp}\alpha)) = \Sigma \backslash \{Female(susan)\}$. □

Remark 4.3.2: Note that in Ex. 3, the operator " $\ominus$ " successfully achieves the contraction of α such that $\Sigma \ominus_\sigma \alpha \not\models \alpha$. This is directly related to the introductory equivalence defined in Sect. 4 in which a necessary condition, in order to verify $\Sigma \models \alpha$, is that α needs to be modeled by every model in Σ, or equivalently $\mathcal{M}(\Sigma) \subseteq \mathcal{M}(\{\alpha\})$.

Example 4 : Let α be the same query used in previous examples such that $\alpha = \exists FRIEND.(Female \sqcap (\exists LOVES.\neg Female))(john)$, and let Σ be a knowledge base such that,

$$\Sigma = \left\{ \begin{array}{l} FRIEND(john, susan) \\ FRIEND(john, andrea) \\ FRIEND(john, bill) \\ LOVES(susan, andrea) \\ LOVES(andrea, bill) \\ LOVES(andrea, john) \\ LOVES(susan, bill) \\ LOVES(bill, john) \\ LOVES(bill, andrea) \\ LOVES(susan, john) \\ Female(susan) \\ \neg Female(john) \end{array} \right\}$$



The following are the interpretation sets related to each model $\mathcal{M}_i$:

[2]Note that the incision made does not relay on any epistemic condition and has being arbitrarily taken.

$\mathcal{M}_1)\ \ Female^{\mathcal{I}} = \{andrea, bill\}, \neg Female^{\mathcal{I}} = \{\}$ $\quad \mathcal{M}_2)\ Female^{\mathcal{I}} = \{bill\}, \neg Female^{\mathcal{I}} = \{andrea\}$
$\mathcal{M}_3)\ \ Female^{\mathcal{I}} = \{andrea\}, \neg Female^{\mathcal{I}} = \{bill\}$ $\quad \mathcal{M}_4)\ Female^{\mathcal{I}} = \{\}, \neg Female^{\mathcal{I}} = \{andrea, bill\}$

Finally, the extended α-kernels ${\Sigma_i}^{\perp\!\!\!\perp}\alpha$ in each model $\mathcal{M}_i$, would be:

${\Sigma_1}^{\perp\!\!\!\perp}\alpha$

$$\left\{ \begin{array}{l} FRIEND(john, susan), Female(susan), \\ LOVES(susan, john), \neg Female(john) \end{array} \right\}$$
$$\left\{ \begin{array}{l} FRIEND(john, andrea), Female(andrea), \\ LOVES(andrea, john), \neg Female(john) \end{array} \right\}$$
$$\left\{ \begin{array}{l} FRIEND(john, bill), Female(bill), \\ LOVES(bill, john), \neg Female(john) \end{array} \right\}$$

${\Sigma_2}^{\perp\!\!\!\perp}\alpha$

$$\left\{ \begin{array}{l} FRIEND(john, susan), Female(susan), \\ LOVES(susan, john), \neg Female(john) \end{array} \right\}$$
$$\left\{ \begin{array}{l} FRIEND(john, bill), Female(bill), \\ LOVES(bill, john), \neg Female(john) \end{array} \right\}$$
$$\left\{ \begin{array}{l} FRIEND(john, bill), Female(bill), \\ LOVES(bill, andrea), \neg Female(andrea) \end{array} \right\}$$
$$\left\{ \begin{array}{l} FRIEND(john, susan), Female(susan), \\ LOVES(susan, andrea), \neg Female(andrea) \end{array} \right\}$$

${\Sigma_3}^{\perp\!\!\!\perp}\alpha$

$$\left\{ \begin{array}{l} FRIEND(john, susan), Female(susan), \\ LOVES(susan, john), \neg Female(john) \end{array} \right\}$$
$$\left\{ \begin{array}{l} FRIEND(john, susan), Female(susan), \\ LOVES(susan, bill), \neg Female(bill) \end{array} \right\}$$
$$\left\{ \begin{array}{l} FRIEND(john, andrea), Female(andrea), \\ LOVES(andrea, bill), \neg Female(bill) \end{array} \right\}$$
$$\left\{ \begin{array}{l} FRIEND(john, andrea), Female(andrea), \\ LOVES(andrea, john), \neg Female(john) \end{array} \right\}$$

${\Sigma_4}^{\perp\!\!\!\perp}\alpha$

$$\left\{ \begin{array}{l} FRIEND(john, susan), Female(susan), \\ LOVES(susan, john), \neg Female(john) \end{array} \right\}$$
$$\left\{ \begin{array}{l} FRIEND(john, susan), Female(susan), \\ LOVES(susan, andrea), \neg Female(andrea) \end{array} \right\}$$
$$\left\{ \begin{array}{l} FRIEND(john, susan), Female(susan), \\ LOVES(susan, bill), \neg Female(bill) \end{array} \right\}$$

A *model selection function* would select one of the considered base extensions in order to cut beliefs from each α-proof in it contained by applying a *model incision function* "σ". In such a case, let consider $\rho(\Sigma^{\perp\!\!\!\perp}\alpha) = {\Sigma_4}^{\perp\!\!\!\perp}\alpha$ as the selected i^{th} set of extended α-kernels, then the *model incision function determined by "ρ"* would be $\sigma(\rho(\Sigma^{\perp\!\!\!\perp}\alpha)) = \sigma({\Sigma_4}^{\perp\!\!\!\perp}\alpha) = \{Female(susan)\}$. [3] Finally, the application of a *model contraction* would be $\Sigma \ominus_{\sigma} \alpha = \Sigma \backslash \{Female(susan)\}$. □

Remark 4.3.3: In the Ex. 4 it is shown that if the base deduces by itself α, *i.e.*, that there exists an *effective α-proof*, therefore the respective proof would be part of every set of extended kernels.

4.4 Anti-Shielding Model Contraction

The proposal of a contraction operation not verifying the success postulate is discussed in several works, in order to analyze the success of a model contraction, let first propose the success and inclusion postulates, inspired in the AGM postulates for contractions originally defined in [1].

(Success) If $\nvdash \alpha$ then $\Sigma \ominus \alpha \nvdash \alpha$
(Inclusion) $\Sigma \ominus \alpha \subseteq \Sigma$

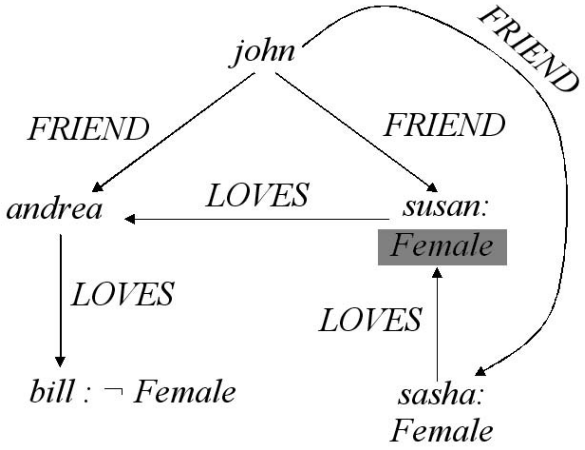
Although the success and inclusion postulates have being verified in previous examples, this is not always possible. Moreover, while inclusion is always verified by a model contraction as exposed before, success does not. In a model contraction as previously specified, some information in a knowledge base may generate the non-satisfiability of the *success postulate*. This information is

[3] As seen before, we adopt an incision which minimizes the quantity of beliefs to be cut off.

being referred as *shielding set* [4] since it represents an epistemic state supporting (in "*background*") an inference of α even after its model contraction. This is clearly shown by the following example:

Example 5 : Let Σ be a knowledge base as exposed below, and α be the same query used in Ex. 1 such that $\alpha = \exists FRIEND.(Female \sqcap (\exists LOVES.\neg Female))(john)$.

$$\Sigma = \left\{ \begin{array}{l} FRIEND(john, susan) \\ FRIEND(john, andrea) \\ FRIEND(john, sasha) \\ LOVES(susan, andrea) \\ LOVES(andrea, bill) \\ LOVES(sasha, susan) \\ Female(susan) \\ \neg Female(bill) \\ Female(sasha) \end{array} \right\}$$



Therefore, the extended α-kernels ${\Sigma_i}^{\perp\!\!\!\perp}\alpha$ would be:

$\mathcal{M}_1$) Model $\mathcal{M}_1$ is such that $\neg Female^{\mathcal{I}} = \{andrea\}, Female^{\mathcal{I}} = \emptyset$

$${\Sigma_1}^{\perp\!\!\!\perp}\alpha = \left\{ \left\{ \begin{array}{l} FRIEND(john, susan), Female(susan), \\ LOVES(susan, andrea), \neg Female(andrea) \end{array} \right\} \right\}$$

$\mathcal{M}_2$) Model $\mathcal{M}_2$ is such that $\neg Female^{\mathcal{I}} = \emptyset, Female^{\mathcal{I}} = \{andrea\}$

$${\Sigma_2}^{\perp\!\!\!\perp}\alpha = \left\{ \left\{ \begin{array}{l} FRIEND(john, andrea), Female(andrea), \\ LOVES(andrea, bill), \neg Female(bill) \end{array} \right\} \right\}$$

A *model selection function* ρ may solve the selection as $\rho(\Sigma^{\perp\!\!\!\perp}\alpha) = {\Sigma_1}^{\perp\!\!\!\perp}\alpha$, then the *model incision function "σ" determined by "ρ"* would be $\sigma(\rho(\Sigma^{\perp\!\!\!\perp}\alpha)) = \sigma({\Sigma_1}^{\perp\!\!\!\perp}\alpha) = \{Female(susan)\}$, and therefore, the application of a *model contraction* would result as $\Sigma \ominus_{\sigma} \alpha = \Sigma \backslash \{Female(susan)\}$.

Note that the success postulate does not hold due to the existence of a *shielding set* H:

$$H = \{FRIEND(john, sasha), Female(sasha), LOVES(sasha, susan)\}$$

where $H \cup \{\neg Female(susan)\} \vdash \alpha$, such that $\neg Female(susan)$ is, after the application of the model contraction, a new assumed belief that helps to verify the query α, and as seen before, its opposite $Female(susan)$ also helps to verify the same query α. So by the application of the contraction as before, we have just generated a new possible world, and afterwards the original query applied to the resultant knowledge base $\Sigma \ominus \alpha \models \alpha$ does still hold. □

Note that although the success postulate does not hold, there is a "weak" version always verified:

(Weak-Success) If $\nvdash \alpha$ then $\Sigma \ominus \alpha \nvdash \alpha$

Observation 4.4.1: A *model contraction* operator " $\ominus$ " determined by a *model incision function* "σ" does verify *weak-success*.
Proof:

[4] The notion of *shielding* was borrowed from [3], where a contraction operator not verifying success was proposed.

Two possibilities arise relaying on the existence of effective α-proofs in Σ. Suppose there is no effective α-proof then from Obs. 4.3.3, $\Sigma \nvdash \alpha$ holds, and since the classic consequence "$\vdash$" verifies *monotony* and " $\ominus$ " verifies *inclusion* it follows that $\Sigma \ominus \alpha \nvdash \alpha$. For the other case in which we suppose that there exists an effective α-proof K in Σ, such that $K \in \rho(\Sigma^{\perp\!\!\!\perp}\alpha)$, it follows that $K_{\mathcal{M}} = \emptyset$ for $K_{\mathcal{M}} \subseteq K$ and therefore $K = K_{\Sigma}$. Hence, from Def. 4.2.2 an incision function "σ" verifies $K_{\Sigma} \cap \sigma(\rho(\Sigma^{\perp\!\!\!\perp}\alpha)) \neq \emptyset$, and since $\Sigma \ominus \alpha = \Sigma \backslash \sigma(\rho(\Sigma^{\perp\!\!\!\perp}\alpha))$ it follows that $K_{\Sigma} \not\subseteq \Sigma \ominus \alpha$. Finally $\Sigma \ominus \alpha$ has no effective α-proof and therefore $\Sigma \ominus \alpha \nvdash \alpha$. ■

The latter example motivates the proposal of a new postulate in order to avoid any *shielding set* in the resultant knowledge base $\Sigma \ominus \alpha$.

(Anti-Shielding) If $\beta \in \Sigma$ and $\beta \notin \Sigma \ominus \alpha$ then $H \cup \{\neg\beta\} \not\models \alpha$ for any $H \subseteq \Sigma \ominus \alpha$

Definition 4.4.2 - Anti-Shielding Model Contraction: Let " $\ominus$ " be a model contraction operator satisfying the *anti-shielding*, then it is referred as *anti-shielding model contraction operator*.

The latter postulate may not only be a property satisfied by some model contractions, this means that by considering the anti-shielding postulate at the time the incision function is being applied, we could always achieve an anti-shielding model contraction operator " $\ominus$ ". A model contraction satisfying anti-shielding is always a desirable operator due to the following observation:

Observation 4.4.3: An *anti-shielding model contraction* operator " $\ominus$ " does verify *success*.
Proof:

If " $\ominus$ " verifies *anti-shielding* it follows that for any $\beta \in \Sigma$ and $\beta \notin \Sigma \ominus \alpha$ then $H \cup \{\neg\beta\} \not\models \alpha$ for any $H \subseteq \Sigma \ominus \alpha$. This proof is shown by supposing to the contrary that *success* is not verified (*i.e.*, $\Sigma \ominus \alpha \models \alpha$).

By Obs. 4.4.1 we know that any model contraction operator does always verify *weak-success* (*i.e.*, $\Sigma \ominus \alpha \nvdash \alpha$), and hence no effective α-proof exists in $\Sigma \ominus \alpha$. This means that there are only possible α-proofs in $\Sigma \ominus \alpha$, thus, we have at least two models $\mathcal{M}_i$ and $\mathcal{M}_j$ with assumed beliefs $\varphi \in \mathcal{M}_i$ and $\neg\varphi \in \mathcal{M}_j$. Therefore, there exist at least two possible α-proofs $K \in ((\Sigma \ominus \alpha)_i{}^{\perp\!\!\!\perp}\alpha)$ and $K' \in ((\Sigma \ominus \alpha)_j{}^{\perp\!\!\!\perp}\alpha)$ such that $K = K_{(\Sigma \ominus \alpha)} \cup K_{\mathcal{M}_i}$ and $K' = K'_{(\Sigma \ominus \alpha)} \cup K'_{\mathcal{M}_j}$, where $\varphi \in K_{\mathcal{M}_i}$ and $\neg\varphi \in K'_{\mathcal{M}_j}$.

This situation is always captured by a model contraction operator, this means that φ was not an assumed belief in Σ (*i.e.*, $\varphi \in \Sigma$), and it has being cut off such that $\varphi \notin \Sigma \ominus \alpha$. Moreover, since $K'_{(\Sigma \ominus \alpha)} \cup K'_{\mathcal{M}_j} \vdash \alpha$ and $\neg\varphi \in K'_{\mathcal{M}_j}$, we have that $K'_{(\Sigma \ominus \alpha)} \cup \{\neg\varphi\} \models \alpha$. Note that assuming $\beta = \varphi$ and $H = K'_{(\Sigma \ominus \alpha)}$ we achieve the absurd $H \cup \{\neg\beta\} \models \alpha$, contradicting *anti-shielding*.

Finally, *success* is verified (*i.e.*, $\Sigma \ominus \alpha \not\models \alpha$). ■

5 Conclusions & Future Work

Some of the theory change classic definitions and postulates exposed in [1] have being generalized in [4] in order to match extra-classic logics like DLs. Considering DLs reasoning services like tableau-based algorithms to solve satisfiability, not only set us up in a more direct theory formalization, but also allow us to work purely description languages with no need to translate beliefs to fragments of first order logic, as we have done before in [7]. Tableaux algorithms are nowadays probably the most important reasoning algorithms used in the area. A distinctive feature of this reasoning service is the way it reasons about incomplete information, inferring new beliefs by combining assumed

ones (from the generated models) and effective knowledge (from the base). All this happens while proving clauses' satisfiability, and indeed, while contracting a sentence from the base. By this, we have a completely different way of reasoning about knowledge due to a multiple generation of base extensions. This all is what motivates the definition of a model contraction and its several components.

Basically, a model contraction is a kind of kernel contraction reinforced in a way such that possible worlds are considered when proving knowledge satisfiability. In this sense, the set of deductions made by this type of reasoning service exceeds the traditional deductions made in a knowledge base. Therefore, a set of kernels in a model contraction (namely extended set of kernels) should consider assumed information, *i.e.*, those beliefs that are part of some possible worlds only. Moreover, an incision function should cut beliefs off in a given model in order to break a proof, for this matter a selection function is defined in order to decide which model is preferable to be incised. Finally, a contraction eliminates those incised beliefs from the knowledge base thus achieving the elimination of the proof in some possible world. Hence, the sentence at issue is no longer verified.

A first approach to some model contraction postulates is given, concentrating this investigation mostly on the success postulate. Here a new type of information sets is recognized as *shielding*, which is a kind of *"protected proof"* for a sentence α standing in *"background"*. Thereafter, a special postulate is proposed in order to deal with this tradeoff, hence achieving success by the model contraction.

As said before, kernel contractions seem to be a special case of model contractions where every required proof is an effective-α proof. In this sense, a deeper investigation is being taking over in order to give a formal characterization of model contractions by means of kernels contractions, relating postulates and axiomatic representations.

References

[1] C. Alchourrón, P. Gärdenfors, and D. Makinson. *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. The Journal of Symbolic Logic*, 50:510–530, 1985.

[2] F. Baader and W. Nutt. Basic Description Logics. *In the Description Logic Handbook, Cambridge University Press*, pages 47–100, 2002.

[3] E. Fermé and S. O. Hansson. Shielded Contraction. *In M-A Williams and H. Rott eds. Frontiers in Belief Revision. Applied Logic Series 22*, pages 85–107, 2001.

[4] G. Flouris. On Belief Change and Ontology Evolution. *Doctoral Dissertation, Department of Computer Science, University of Crete*, February 2006.

[5] E. Franconi. Propositional Description Logics. *From his course Description Logics dictated during argentinean springtime at Universidad Nacional del Sur, Bahía Blanca*, 2006.

[6] S. O. Hansson. Kernel Contraction. *The Journal of Symbolic Logic*, 59:845–859, 1994.

[7] M. Moguillansky and M. Falappa. A Non-monotonic Description Logics Model for Merging Terminologies. *Revista Iberoamericana de Inteligencia Artificial (AEPIA), ISSN 1137-3601*, 2007. at press.

[8] M. Schmidt-Schauß and G. Smolka. Attributive Concept Descriptions with Complements. *Artificial Intelligence*, 48(1):1–26, 1991.

A Preliminary Investigation on a Revision-Based Approach to the Status of Warrant*

Martín O. Moguillansky Nicolás D. Rotstein
Marcelo A. Falappa Guillermo R. Simari

Consejo de Investigaciones Científicas y Técnicas (CONICET)
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de la Computación (DCIC)
Universidad Nacional del Sur (UNS)
Av. Alem 1253 - (B8000CPB) Bahía Blanca - Argentina
PHONE/FAX: (+54)(291)459-5136
E-MAIL: {mom, ndr, mfalappa, grs}@cs.uns.edu.ar

Abstract

In this article we are presenting a new perspective on the matter of belief revision by its relation to argumentation systems. Our approach is based on the argumentative formalism Defeasible Logic Programming, and therefore we propose a revision of a defeasible logic program by an argument. The revision operators here introduced are defined as prioritized, since they ensure warrant of the conclusion of the argument being added to the program following a particular minimal change principle. To achieve this, we give two different approaches: one regarding arguments in the classical sense, whereas the other considers the revision by arguments that also include strict rules and facts. Finally, a brief discussion about the relation between our approach and the basic theory of belief revision is exposed, along with a description of other possible minimal change principles.

Keywords: belief revision, argumentation, defeasible logic programming, non-monotonic reasoning.

1 Introduction & Motivation

This work offers a first approach to revision in argumentation systems, using Defeasible Logic Programming (DELP) as base formalism. The objective is to define an **argument revision operator** that ensures warrant of the conclusion of the (external) argument being added to a program. In that sense, this operator will be **prioritized**. When we revise a program $\mathcal{P}$ by an argument $\langle \mathcal{A}, \alpha \rangle$, the program resulting from the revision will be such that $\mathcal{A}$ is an undefeated argument and α is then warranted. Because of this, we named the operator **warrant-prioritized revision operator**.

*This article assumes background knowledge on argumentation and belief revision from the reader.

Partially financed by CONICET (PIP 5050), Universidad Nacional del Sur (PGI 24/ZN11) and Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096).

The main issue underlying warrant-prioritized argument revision lies in the selection of arguments and the incisions that have to be made over them. Incisions will make these arguments "disappear", but selections have to be done carefully, following some minimal change principle. In this work, we present one minimal change principle, thus defining a warrant-prioritized revision operator. The corresponding selection and incision functions are defined, along with some properties it should verify. This revision operator is therefore reconsidered, by applying it to extended arguments.

The article is structured as follows: Section 2 gives an overview of the main concepts involved in the DELP formalism, Section 3 describes the notions of the belief revision theory we inspired from to define this approach, Section 4 explains in detail the two versions of the revision operator, and Section 5 gives a final discussion on revision in argument systems and poses future lines of work.

2 DeLP Overview

Defeasible Logic Programming (DELP) combines results of Logic Programming and Defeasible Argumentation. The system is fully implemented and is available online [1]. A brief explanation is included below (see [6] for full details). A DELP-program $\mathcal{P}$ is a set of facts, strict rules and defeasible rules. *Facts* are ground literals representing atomic information or the negation of atomic information using the strong negation "$\sim$" (*e.g.*, $chicken(little)$ or $\sim scared(little)$). *Strict Rules* represent non-defeasible information and are denoted $L_0 \leftarrow L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals (*e.g.*, $bird \leftarrow chicken$) or $\sim innocent \leftarrow guilty$). *Defeasible Rules* represent tentative information and are denoted $L_0 -\!\prec L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals (*e.g.*, $\sim flies -\!\prec chicken$ or $flies -\!\prec chicken, scared$).

When required, $\mathcal{P}$ is denoted (Π, Δ) distinguishing the subset Π of facts and strict rules, and the subset Δ of defeasible rules (see Ex. 1). *Strong negation* is allowed in the head of rules, and hence may be used to represent contradictory knowledge. From a program (Π, Δ) contradictory literals could be derived. Nevertheless, the set Π (which is used to represent non-defeasible information) must possess certain internal coherence, *i.e.*, no pair of contradictory literals can be derived from Π.

A defeasible rule is used to represent tentative information that may be used if nothing could be posed against it. Observe that strict and defeasible rules are ground. However, following the usual convention [12], some examples use "schematic rules" with variables. To distinguish variables, as usual, they start with an uppercase letter.

Example 1 *Consider the* DELP*-program* (Π_1, Δ_1) *where:*

$$\Pi_1 = \left\{ \begin{array}{ll} (bird(X) \leftarrow chicken(X)) & chicken(little) \\ chicken(tina) & scared(tina) \end{array} \right\}$$

$$\Delta_1 = \left\{ \begin{array}{l} flies(X) -\!\prec bird(X) \\ flies(X) -\!\prec chicken(X), scared(X) \\ \sim flies(X) -\!\prec chicken(X) \end{array} \right\}$$

This program has three defeasible rules representing tentative information about the flying ability of birds in general, and about regular chickens and scared ones. It also has a strict rule expressing that every chicken is a bird, and three facts: 'tina' and 'little' are chickens, and 'tina' is scared.

From a program is possible to derive contradictory literals, *e.g.*, from (Π_1, Δ_1) of Ex. 1 it is possible to derive $flies(tina)$ and $\sim flies(tina)$. For the treatment of contradictory knowledge DELP incorporates a defeasible argumentation formalism. This formalism allows the identification of the pieces of knowledge that are in contradiction, and a *dialectical process* is used for deciding which information prevails as warranted. This dialectical process (see below) involves the construction and

evaluation of arguments that either support or interfere with the query under analysis. As we will show next, arguments supporting the answer for a given query will be shown in a particular way using *dialectical trees*. The definition of dialectical tree will be included below, but first, we will give a brief explanation of other related concepts (for the details see [6]).

Definition 1 (Argument Structure) *Let* (Π, Δ) *be a* DELP*-program,* $\langle \mathcal{A}, \alpha \rangle$ *is an argument structure for a literal* α *from* (Π, Δ)*, if* $\mathcal{A}$ *is the minimal set of defeasible rules (*$\mathcal{A} \subseteq \Delta$*), such that: (1) there exists a defeasible derivation for* α *from* $\Pi \cup \mathcal{A}$*, and (2) the set* $\Pi \cup \mathcal{A}$ *is non-contradictory.*

Example 2 *From the* DELP*-program* (Π_1, Δ_1) *the following arguments can be obtained:*
$\langle \mathcal{A}_1, flies(tina) \rangle = \langle \{ flies(tina) \text{–}\prec bird(tina) \}, flies(tina) \rangle$
$\langle \mathcal{A}_2, \sim flies(tina) \rangle = \langle \{ \sim flies(tina) \text{–}\prec chicken(tina) \}, \sim flies(tina) \rangle$
$\langle \mathcal{A}_3, flies(tina) \rangle = \langle \{ flies(tina) \text{–}\prec chicken(tina), scared(tina) \}, flies(tina) \rangle$

A literal L is *warranted* if there exists a non-defeated argument $\mathcal{A}$ supporting L. To establish if $\langle \mathcal{A}, \alpha \rangle$ is a non-defeated argument, *defeaters* for $\langle \mathcal{A}, \alpha \rangle$ are considered, *i.e.*, counter-arguments that by some criterion are preferred to $\langle \mathcal{A}, \alpha \rangle$. In DELP, the comparison criterion is usually *generalized specificity*, but in the examples given in this paper we will abstract away this criterion, since in this work it introduces unnecessary complications. Since defeaters are arguments, there may exist defeaters for them, and defeaters for these defeaters, and so on. Thus, a sequence of arguments called *dialectical line* is constructed, where each argument defeats its predecessor. To avoid undesirable sequences, that may represent circular or fallacious argumentation lines, in DELP a dialectical line is *acceptable* if it satisfies certain constraints (see [6]).

Example 3 *From Ex. 2, we have that argument* $\langle \mathcal{A}_2, \sim flies(tina) \rangle$ *defeats* $\langle \mathcal{A}_1, flies(tina) \rangle$*, argument* $\langle \mathcal{A}_3, flies(tina) \rangle$ *is a defeater for* $\langle \mathcal{A}_2, \sim flies(tina) \rangle$*, and the arguments sequence* $[\langle \mathcal{A}_1, flies(tina) \rangle, \langle \mathcal{A}_2, \sim flies(tina) \rangle, \langle \mathcal{A}_3, flies(tina) \rangle]$ *is an acceptable argumentation line.*

Clearly, there might be more than one defeater for a particular argument. Therefore, many acceptable argumentation lines could arise from one argument, leading to a tree structure.

Definition 2 (Dialectical tree [6]) *A dialectical tree for* $\langle \mathcal{A}_0, h_0 \rangle$*, denoted* $\mathcal{T}(\langle \mathcal{A}_0, h_0 \rangle)$*, is defined as follows:*
(1) The root of the tree is labelled with $\langle \mathcal{A}_0, h_0 \rangle$*.*
(2) Let N *be a node of the tree labelled* $\langle \mathcal{A}_n, h_n \rangle$*, and*
$\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle]$ *be the sequence of labels of the path from the root to* N*.*
Let $\{ \langle \mathcal{B}_1, q_1 \rangle, \langle \mathcal{B}_2, q_2 \rangle, \ldots, \langle \mathcal{B}_k, q_k \rangle \}$ *be the set of all the defeaters for* $\langle \mathcal{A}_n, h_n \rangle$*. For each defeater* $\langle \mathcal{B}_i, q_i \rangle$ $(1 \leq i \leq k)$*, such that, the argumentation line* $\Lambda' = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle, \langle \mathcal{B}_i, q_i \rangle]$ *is acceptable, then the node* N *has a child* N_i *labelled* $\langle \mathcal{B}_i, q_i \rangle$*.*
If there is no defeater for $\langle \mathcal{A}_n, h_n \rangle$ *or there is no* $\langle \mathcal{B}_i, q_i \rangle$ *such that* Λ' *is acceptable, then* N *is a leaf.*

In a dialectical tree, every node (except the root) represents a defeater of its parent, and leaves correspond to non-defeated arguments. Each path from the root to a leaf corresponds to a different acceptable argumentation line. A dialectical tree provides a structure for considering all the possible acceptable argumentation lines that can be generated for deciding whether an argument is defeated. We call this tree *dialectical* because it represents an exhaustive dialectical analysis for the argument in its root.

Given a literal h and an argument $\langle \mathcal{A}, h \rangle$ from a program $\mathcal{P}$, to decide whether a literal h is warranted, every node in the dialectical tree $\mathcal{T}(\langle \mathcal{A}, h \rangle)$ is recursively marked as "D" (*defeated*) or "U" (*undefeated*), obtaining a marked dialectical tree $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h \rangle)$ as follows:

1. All leaves in $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h\rangle)$ are marked as "U"s, and
2. Let $\langle \mathcal{B}, q\rangle$ be an inner node of $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h\rangle)$. Then $\langle \mathcal{B}, q\rangle$ will be marked as "U" iff every child of $\langle \mathcal{B}, q\rangle$ is marked as "D". The node $\langle \mathcal{B}, q\rangle$ will be marked as "D" iff it has at least a child marked as "U".

Given an argument $\langle \mathcal{A}, h\rangle$ obtained from $\mathcal{P}$, if the root of $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h\rangle)$ is marked as "U", then we will say that $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h\rangle)$ *warrants* h and that h is *warranted* from $\mathcal{P}$.

In this paper, marked dialectical trees will be depicted as a tree of triangles where edges denote the defeat relation (in Figure 1, three marked dialectical trees are shown). An argument $\langle \mathcal{A}, h\rangle$ will be depicted as a triangle, where its upper vertex is labelled with the conclusion h, and the set of defeasible rules $\mathcal{A}$ are associated with the triangle itself. Gray triangles will be undefeated arguments, whereas white triangles will depict defeated arguments. In the rest of the article we will refer to marked dialectical trees just as "D-tree".

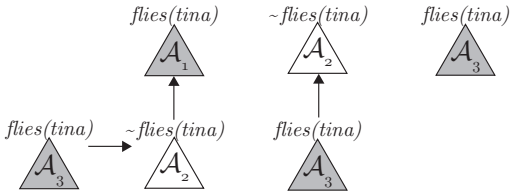


Figure 1: D-trees for $flies(tina)$

Example 4 *(Extends Ex. 3) Figure 1 shows the D-tree for $\mathcal{T}^*_{\mathcal{P}_1}(\langle \mathcal{A}_1, f\rangle)$ (the leftmost tree), which has only one argumentation line. Observe that the argument $\langle \mathcal{A}_2, \sim f\rangle$ interferes with the warrant of 'flies(tina)' and the argument $\langle \mathcal{A}_3, f\rangle$ reinstates $\langle \mathcal{A}_1, f\rangle$. The root of $\mathcal{T}^*_{\mathcal{P}_1}(\langle \mathcal{A}_1, f\rangle)$ is marked as "U" and therefore the literal 'flies(tina)' is warranted.*

3 Belief Revision Overview

A **belief base** is a knowledge state represented by a set of sentences not necessarily closed under logical consequence. A **belief set** is a set of sentences in a given language, closed under logical consequence. In general, a belief set is infinite being this the main reason of the impossibility to deal with this kind of sets in a computer. Instead, it is possible to characterize the properties that each of the change operations should satisfy on any finite representation of a knowledge state.

Classic operations in the theory change [2] are known as expansions, contractions, and revisions. An **Expansion** operation noted with "$+$", adds a new belief to the epistemic state, without guaranteeing its consistency after the operation. A **Contraction** operation, noted with "$-$", eliminates a belief α from the epistemic state and those beliefs that make possible its deduction or inference. The sentences to eliminate might represent the *minimal change* on the epistemic state. Finally, a **Revision** operation ("$*$") inserts sentences to the epistemic state, guaranteeing consistency (if it was consistent before the operation). This means that a revision adds a new belief and perhaps it eliminates other ones in order to avoid inconsistencies.

Other non-classical operations exists, like **Merge** [13] noted with " $\circ$ ", which fusions belief bases or sets assuring a consistent resultant epistemic state, and **Consolidations** (" ! ") that restore consistency to a contradictory epistemic state. Usually, slight extensions or modifications of these operations are needed in order to capture different improved features of the environment required to work

in. This is the case of the operator " $\diamond$ " used to represent a **Kernel Revision by a Set of Sentences** [4] operation, which defines a non-prioritized version of a kernel revision operation, providing the knowledge to be revised and its logical proof (or argument) given by a consistent set of sentences.

3.1 Kernel Contractions

The **Kernel Contraction** operator, applicable to belief bases and belief sets, consists of a contraction operator capable of the selection and elimination of those beliefs in K that contribute to infer α.

Definition 3 (Set of Kernels [9]) *Let K be a set of sentences and α a sentence. The set $K^{\perp\!\!\!\perp}\alpha$, called* set of kernels *is the set of sets K' such that* (1) $K' \subseteq K$, (2) $K' \vdash \alpha$, *and* (3) *if $K'' \subset K'$ then $K'' \nvdash \alpha$. The set $K^{\perp\!\!\!\perp}\alpha$ is also called* set of α-kernels *and each one of its elements are called* α-kernel.

For the success of a contraction operation we need to eliminate, at least, one element of each α-kernel. The elements to be eliminated are selected by an **Incision Function**.

Definition 4 (Incision Function [9]) *Let K be a set of sentences and "σ" be an* incision function *for it such that for any sentence α it verifies,* (1) $\sigma(K^{\perp\!\!\!\perp}\alpha) \subseteq \bigcup(K^{\perp\!\!\!\perp}\alpha)$ *and* (2) *If $K' \in K^{\perp\!\!\!\perp}\alpha$ and $K' \neq \emptyset$ then $K' \cap \sigma(K^{\perp\!\!\!\perp}\alpha) \neq \emptyset$.*

Once the incision function was applied we must eliminate from K those sentences that the incision has selected, *i.e.*, the new belief base would consist of all those sentences that kept outside of the scope of σ.

Definition 5 (Kernel Contraction Determined by "σ" [9]) *Let K be a set of sentences, α a sentence, and $K^{\perp\!\!\!\perp}\alpha$ the set of α-kernels of K. Let "σ" be an incision function for K. The operator " $-_\sigma$ ", called* kernel contraction determined by "σ", *is defined as, $K -_\sigma \alpha = K \backslash \sigma(K^{\perp\!\!\!\perp}\alpha)$.*

Finally, an operator " $-$ " is a kernel contraction operator for K if and only if there exists an incision function "σ" such that $K - \alpha = K -_\sigma \alpha$ for all sentence α.

3.2 Consistent Incorporation of Beliefs

A revision operator "$*$" looks for the addition of the new belief α to the belief set K, and therefore the assurance that the resulting belief set $K * \alpha$ is consistent (unless α is inconsistent). The first task can be accomplished by expansion by α. The second can be accomplished by prior contraction by its negation $\sim\alpha$. If a belief set does not imply $\sim\alpha$, then α can be added to it without loss of consistency. This composition of sub-operations gives rise to the following definition of a revision operator [7, 11]:

$$\textbf{(Levi Identity)}\ K * \alpha = (K - \sim\alpha) + \alpha$$

We will define the revision operation in a set K regarding a sentence α, by means of the *Levi Identity*, assuming that " $-$ " is a *kernel contraction* operator determined by an incision function "σ".

Definition 6 (Internal (External) Kernel Revision [10]) *Let " $-$ " be a kernel contraction for a set K. Then the **Internal Kernel Revision** operator for K is defined as $K \mp_\sigma \alpha = (K - \sim\alpha) + \alpha$. Analogously, an **External Kernel Revision** operator is defined as $K \pm_\sigma \alpha = (K + \alpha) - \sim\alpha$.*

Finally, a kernel revision operator "$*$" may be characterized by either an internal "$\mp_\sigma$" or an external "$\pm_\sigma$" kernel revision.

3.3 Non Prioritized Revisions

The classic revision operation is characterized by the postulates of rationality introduced by Gärdenfors [8], some of them have been argued for being considered arbitraries. Particularly, the *success* postulate ($\alpha \in K * \alpha$) establishes that a new information to be revised in an epistemic state must be part of it, despite that other beliefs in the agent's state must be eliminated in order to maintain its consistency. For that purpose is interesting to define new types of revision operations to "catch" the information in a more intuitively way such that a new information has "*no absolute priority*" over those in the epistemic state.

Definition 7 (Explanation Set [5]) *The set A is an explanation for α iff it means a **minimal** proof for α, it is consistent and it is not self-explanatory, i.e., $\alpha \nvdash A$.*

Usually one does not totally accept what others inform but only what one considers to be relevant. This property is known as *partial acceptation*, and its behavior may be modeled by a multiple revision operator as follows:

Definition 8 (Non-Prioritized Multiple Revision [5]) *Let "σ" be an incision function, and let K and A be two sets of sentences, such that K is consistent and A finite. The **Non-Prioritized** Kernel Revision by a Set of Sentences operator "$\diamond_{\mathcal{N}}$" is defined as follows:*

$$K \diamond_{\mathcal{N}} A = (K \cup A) \backslash \sigma((K \cup A)^{\perp\!\!\!\perp} \perp)$$

4 Argument Revision Operators

Intuitively, a "**Warrant-Prioritized Argument Revision Operator**" (for short: WP Argument Revision Operator) revises a given program $\mathcal{P} = (\Pi, \Delta)$ by an external argument $\langle \mathcal{A}, \alpha \rangle$. Moreover, this argument ends up being warranted from the program resulting from the revision, provided that $\mathcal{A} \cup \Pi$ has a defeasible derivation for α. The set Π of strict rules and facts represents (in a way) the current state of the world. The external argument $\langle \mathcal{A}, \alpha \rangle$ provides a set of defeasible rules that jointly with the state of the world decides in favor of the conclusion α, *i.e.*, it poses a reason to believe in it. Hence, this argument does not stand by itself, but in conjunction with the strict part of the program it is being added to, *i.e.*, α is defeasibly derived from $\mathcal{A} \cup \Pi$.

Although it would be interesting to revise a program by $\langle \mathcal{A}, \alpha \rangle$ only when α is not already warranted (by another argument), it might be desirable to have $\mathcal{A}$ as an undefeated argument. In our approach, we take this last posture: the WP Argument Revision Operator will ensure $\mathcal{A}$ to be an undefeated argument. In this way, α would be always warranted.

For this matter, a *hypothetical dialectical tree* rooted on $\langle \mathcal{A}, \alpha \rangle$ is built. The D-tree is deemed as *"hypothetical"* due to $\langle \mathcal{A}, \alpha \rangle$ not belonging to $\mathcal{P}$. Incisions over arguments in this tree are made in order to turn $\mathcal{A}$ into an undefeated argument. Selections (consequently, incisions) must agree with some minimal change principle. In this work, we propose a principle that attempts to ensure minimal deletion of the DELP-program rules.

Finally, in the examples given throughout the article, we abstract away the argument comparison criterion: we will just give DELP-programs, pointing out which is the associated D-tree for the argument being added, and thereafter the analysis begins.

4.1 WP Argument Revision – Minimal Change wrt. the DeLP-program

A WP Argument Revision Operator "$*^{\omega}_{\mathcal{P}}$" attempts to insert an argument $\langle \mathcal{A}, \alpha \rangle$ into a program $\mathcal{P}$, in such a way that α turns out to be warranted from $\mathcal{P} *^{\omega}_{\mathcal{P}} \langle \mathcal{A}, \alpha \rangle$. Revising a program $\mathcal{P} = (\Pi, \Delta)$ by an argument $\langle \mathcal{A}, \alpha \rangle$ involves the generation of a hypothetical D-tree rooted in $\langle \mathcal{A}, \alpha \rangle$, namely $\mathcal{T}^{*}_{\mathcal{P}'}(\langle \mathcal{A}, \alpha \rangle)$, where $\mathcal{P}' = (\Pi, \Delta \cup \mathcal{A})$. Therefore, since we want α to be warranted, those undefeated defeaters for $\langle \mathcal{A}, \alpha \rangle$ will be *cut off* in order to turn $\langle \mathcal{A}, \alpha \rangle$ into an undefeated argument.

Definition 9 (Argument Selection Function "$\gamma^{\omega}_{\mathcal{P}}$") *Let $T = \mathcal{T}^{*}_{\mathcal{P}'}(\langle \mathcal{A}, \alpha \rangle)$ be a* D-tree *and λ_i a dialectical line rooted in $\langle \mathcal{A}, \alpha \rangle$, then $\gamma^{\omega}_{\mathcal{P}}(\lambda_i) = \mathcal{B}$ iff $\mathcal{B}$ is a defeater for $\langle \mathcal{A}, \alpha \rangle$ marked as* **undefeated** *in T. From now on, the argument selected in the i^{th} dialectical line will be called Ψ_i.*

In general, selecting defeaters for the root argument ensures a minimal deletion of defeasible rules from the DELP-program at issue. That is because the deletion of a root's defeater eliminates a whole branch. Trying to achieve the same result by deleting rules from "lower" arguments would affect a greater amount of arguments, due to the possibility of branching.

Definition 10 (Argument Incision Function "$\sigma^{\omega}_{\mathcal{P}}$") *Let Ψ_i be the* ***"less relevant interference argument"*** *determined by an argument selection function $\gamma^{\omega}_{\mathcal{P}}$ in the dialectical line λ_i. Then a function $\sigma^{\omega}_{\mathcal{P}}$ is an* argument incision function *iff it verifies $\emptyset \subset \sigma^{\omega}_{\mathcal{P}}(\Psi_i) \subseteq \Psi_i$.*

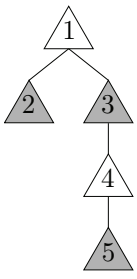
Example 5 *Let consider a program $\mathcal{P}_5 = (\Pi_5, \Delta_5)$, where:*

$$\Pi_5 = \left\{ \begin{array}{l} t, \\ z \end{array} \right\} \quad \Delta_5 = \left\{ \begin{array}{l} \sim a \prec y, y \prec x, \sim a \prec z, \\ a \prec w, w \prec y, \sim a \prec t \end{array} \right\}$$

From $\mathcal{P}_5$ we can build the following arguments:

$\langle 2, \sim a \rangle$: $\langle \{\sim a \prec y, y \prec x, x \prec z\}, \sim a \rangle$
$\langle 3, \sim a \rangle$: $\langle \{\sim a \prec z\}, \sim a \rangle$
$\langle 4, a \rangle$: $\langle \{a \prec w, w \prec y, y \prec x\}, a \rangle$
$\langle 5, \sim a \rangle$: $\langle \{\sim a \prec t\}, \sim a \rangle$

Consider that $\mathcal{P}_5$ is revised by the argument $\langle 1, a \rangle$: $\langle \{a \prec x, x \prec z\}, a \rangle$. From now on, we abstract away both the argument preference criterion and dialectical line acceptability, so we will provide the attacks between arguments with no further discussion: arguments 2 and 3 attack 1, argument 4 attacks 3, and 5 attacks 4. Then, assume that the hypothetical D-tree on the right is built from $\mathcal{P}'_5 = (\Pi_5, \Delta_5 \cup 1)$. Here the argument selection function $\gamma^{\omega}_{\mathcal{P}}$ selects the arguments 2 and 3 to be cut off, while the argument incision function $\sigma^{\omega}_{\mathcal{P}}$ applied over them could be any subset.



To make an argument disappear, an incision over it must be performed. However, that incision might have a *collateral effect* and make another argument/s from the tree disappear. That is, the rules being cut off from an incised argument might belong to more arguments in the tree, and then the impact on the tree structure would be greater.

Definition 11 (Collateral Incision) *Let $\sigma^{\omega}_{\mathcal{P}}(\Psi)$ be an incision and $\mathcal{B}$ be any argument in the tree. If $\sigma^{\omega}_{\mathcal{P}}(\Psi) \cap \mathcal{B} \neq \emptyset$ holds, then $\sigma^{\omega}_{\mathcal{P}}(\Psi) \cap \mathcal{B}$ is called a* ***collateral incision*** *over $\mathcal{B}$.*

The argument incision function should be applied to the portion of the argument that does not belong to the root argument, *i.e.*, *it should avoid any collateral incision over the root argument.* The motivation of this property is that if a rule belonging to the root argument were to be cut off, this argument would no longer hold, turning impossible to warrant its conclusion. Therefore, the following property is proposed for an argument incision function "$\sigma^{\omega}_{\mathcal{P}}$":

(Root-Preservation) $\sigma^{\omega}_{\mathcal{P}}(\Psi_i) \cap \mathcal{A} = \emptyset$, where $\gamma^{\omega}_{\mathcal{P}}(\lambda_i) = \Psi_i$

Example 6 *Let us consider program $\mathcal{P}'_5$ from Ex. 5. Here, the incision over argument 2 could be any subset that does not contain the defeasible rule $x \prec\!\!- z$ (which belongs to the root argument 1), that is, any subset of $\{\sim a \prec\!\!- y, y \prec\!\!- x\}$. For instance, $\sigma^{\omega}_{\mathcal{P}}(2) = \{\sim a \prec\!\!- y\}$. The incision over argument 3, however, must be the single rule it contains: $\sigma^{\omega}_{\mathcal{P}}(3) = \{\sim a \prec\!\!- z\}$. Therefore we have that $\sigma^{\omega}_{\mathcal{P}}$ satisfies root-preservation.*

Remark 1 *Since arguments are minimal, given an argument $\langle \mathcal{B}, \beta \rangle$, it is clear that there is no defeasible derivation for β from $\Pi \cup (\mathcal{B} \setminus \sigma^{\omega}_{\mathcal{P}}(\mathcal{B}))$.*

When a collateral incision arises, some side effects may occur compromising the objective of the revision (*i.e.*, the root argument might end up defeated). This may happen in case a collateral incision affects a supporting argument in a dialectical line which originally had a defeated defeater (for the root), thus yielding it undefeated. This situation is captured by the following remark.

Remark 2 *The marking of a D-tree is considered dynamic, this is, it may change by a collateral effect of the applied incisions. Thereafter, if the status of a dialectical line has changed (now having an undefeated defeater for the root), then it should be further affected by an incision function.*

Definition 12 (Root-Preserving Argument Incision Function) *An argument incision function "$\sigma^{\omega}_{\mathcal{P}}$" determined by an argument selection function "$\gamma^{\omega}_{\mathcal{P}}$" is called **root-preserving argument incision function** if it verifies **root-preservation**.*

Now that both the selection and the incision function are defined, the WP Argument Revision operation can be formally defined.

Definition 13 (WP Argument Revision) *Let $\mathcal{P}$ be a program such that $\mathcal{P} = (\Pi, \Delta)$. A revision operation of $\mathcal{P}$ by an argument $\langle \mathcal{A}, \alpha \rangle$, namely $\mathcal{P} *^{\omega}_{\mathcal{P}} \langle \mathcal{A}, \alpha \rangle$, is defined by means of a **root-preserving argument incision function** "$\sigma^{\omega}_{\mathcal{P}}$" as follows:*

$$\mathcal{P} *^{\omega}_{\mathcal{P}} \langle \mathcal{A}, \alpha \rangle = (\Pi, \mathcal{A} \cup \Delta \setminus \bigcup_i (\sigma^{\omega}_{\mathcal{P}}(\Psi_i)))$$

Theorem 1 *Let $\mathcal{P}_R = \mathcal{P} *^{\omega}_{\mathcal{P}} \langle \mathcal{A}, \alpha \rangle$ be a revised defeasible logic program by an argument $\langle \mathcal{A}, \alpha \rangle$, then α is warranted from $\mathcal{P}_R$.*

Example 7 *From Ex. 6 we have the incisions $\sigma^{\omega}_{\mathcal{P}}(2) = \{\sim a \prec\!\!- y\}$ and $\sigma^{\omega}_{\mathcal{P}}(3) = \{\sim a \prec\!\!- z\}$. Then, from the revision $\mathcal{P}_5 *^{\omega}_{\mathcal{P}} \langle \{a \prec\!\!- x, x \prec\!\!- z\}, a \rangle$ made in Ex. 5 we have:*

$$\mathcal{P}_{5R} = (\Pi_5, \Delta_5 \cup \{a \prec\!\!- x, x \prec\!\!- z\} \setminus \{\sim a \prec\!\!- y, \sim a \prec\!\!- z\}) = \left(\left\{ \begin{array}{l} t, \\ z \end{array} \right\}, \left\{ \begin{array}{l} a \prec\!\!- x,\ y \prec\!\!- x,\ x \prec\!\!- z, \\ a \prec\!\!- w,\ w \prec\!\!- y,\ \sim a \prec\!\!- t \end{array} \right\} \right)$$

From $\mathcal{P}_{5R}$ literal a is warranted, since the D-tree is just the root of the one depicted in Ex. 5.

4.2 WP Argument Revision Considering Extended Arguments

Revising a program (Π, Δ) by an argument $\langle \mathcal{A}, \alpha \rangle$ assuming that $\mathcal{A} \cup \Pi$ derives α might be too restrictive. Then, from the operator explained in the last section, we can consider a variation of it that revises a program by an *extended argument*. These arguments will contain strict rules and facts, besides defeasible rules. This characteristic gives them the possibility of being *self-contained*, in the sense that they derive a conclusion just by themselves. However, extended arguments bring about a main drawback: consistency checking, *i.e.*, when a program is revised by an extended argument, the join of their sets of strict rules must be non-contradictory. Thereafter, this join can be defined following several policies, *i.e.*, deleting rules in contradiction, turn them into defeasible rules, *etc*. Moreover, this policy can be applied either over the strict rules of the program, over the strict rules of the argument, or both. Since we are defining a prioritized argument revision operator, we are going to keep the first option: only strict rules of the program will be affected. Next, we formally define the notion of extended argument.

Definition 14 (Extended Argument) *Given a Π set of strict rules and a set Δ of defeasible rules, a pair $\langle (\Pi, \Delta), \alpha \rangle$ is an extended argument structure for a literal α, if there is a minimal defeasible derivation for α from $\Pi \cup \Delta$, and $\Pi \cup \Delta$ is non-contradictory.*

Every concept from the DELP theory (such as (marked) dialectical trees) can be translated by replacing the classical notion of argument. A redefinition of these concepts is not to be done in this article, due to space reasons and because they are very similar to the original ones.

An extended-argument revision operation is noted as $\mathcal{P} \circledast^{\omega}_{\mathcal{P}} \langle (\Pi', \Delta'), \alpha \rangle$, where $\mathcal{P} = (\Pi, \Delta)$. Then, we should consistently join both strict sets Π and Π', by means of a *prioritized* multiple revision operator, namely "$\diamond_{\mathcal{P}}$", consequently defined as:

Definition 15 (Prioritized Multiple Revision) *Let "σ" be an incision function (Def. 4), K and A be two sets of sentences, such that K is consistent and A finite, and let $K^{\perp\!\!\!\perp}\beta$ be the set of β-kernels (Def. 3), such that $A \vdash \sim\beta$. The **Prioritized** Kernel Revision by a Set of Sentences operator "$\diamond_{\mathcal{P}}$" is defined as follows:*

$$K \diamond_{\mathcal{P}} A = (K \backslash \sigma(K^{\perp\!\!\!\perp}\beta)) \cup A$$

Note that the definition of "$\diamond_{\mathcal{P}}$" is inspired by the theory proposed in [5] and mostly by the definition of its non-prioritized version "$\diamond_{\mathcal{N}}$". The following properties are verified.

Proposition 1 *Given a **multiple prioritized revision** operation $K \diamond_{\mathcal{P}} A$, the following properties hold:*

1. $A \subseteq K \diamond_{\mathcal{P}} A$
2. $K \not\subseteq K \diamond_{\mathcal{P}} A$ *iff* $K \cup A \vdash \bot$

Therefore, as part of the definition of a *Warrant-Prioritized Extended-Argument Revision Operation* $\mathcal{P} \circledast^{\omega}_{\mathcal{P}} \langle (\Pi', \Delta'), \alpha \rangle$ (where $\mathcal{P} = (\Pi, \Delta)$), we should achieve the consistent joint of both strict sets of rules Π and Π', such that $\Pi \diamond_{\mathcal{P}} \Pi'$. Afterwards, the sets of defeasible rules Δ and Δ' should be joined. This may be easily achieved since there is no need to preserve consistence by a set of defeasible rules, then a preliminary version might be just $\Delta \cup \Delta'$. But furthermore, in order to preserve beliefs, a slight modification is proposed by adopting the policy of "*weakening*" the erased strict rules ρ, selected from Π by an incision function σ. Finally, while $\rho \in \sigma(\Pi^{\perp\!\!\!\perp}\beta)$ or equivalently $\rho \in \Pi \backslash \Pi \diamond_{\mathcal{P}} \Pi'$, the referred "*weakening*" is performed by means of a function δ such that $\delta(\rho)$ is the defeasible version of the strict rule ρ. This idea is originally exposed in [5], where a first approach

of revision in argumentative systems is given, and also in [14], where at one stage of the architecture two DELP-programs have to be combined.

Supposing that the operator "$\circledast^{\omega}_{\mathcal{P}}$" would define a new program $\mathcal{P}_R = (\Pi \diamond_{\mathcal{P}} \Pi', \Delta \cup \delta(\Pi \backslash \Pi \diamond_{\mathcal{P}} \Pi') \cup \Delta')$, it could not be possible to ensure that α is warranted from $\mathcal{P}_R$. Therefore, in order to achieve warrant for α, we propose to define the operator "$\circledast^{\omega}_{\mathcal{P}}$" by means of the operator "$*^{\omega}_{\mathcal{P}}$" previously defined, as follows:

Definition 16 (WP Extended-Argument Revision) *A Warrant-Prioritized Extended-Argument Revision Operator "$\circledast^{\omega}_{\mathcal{P}}$" is defined in terms of the operator "$\diamond_{\mathcal{P}}$" and the WP Argument Revision operator "$*^{\omega}_{\mathcal{P}}$" as follows:*

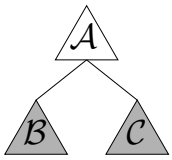
$$(\Pi, \Delta) \circledast^{\omega}_{\mathcal{P}} \langle (\Pi', \Delta'), \alpha \rangle = (\Pi \diamond_{\mathcal{P}} \Pi', \Delta \cup \delta(\Pi \backslash \Pi \diamond_{\mathcal{P}} \Pi')) *^{\omega}_{\mathcal{P}} \langle \Delta', \alpha \rangle$$

Example 8 *Consider the extended argument $\mathcal{A} = \langle (\{x \leftarrow t, t\}, \{a \prec\!\!\!- x\}), a \rangle$ and the* DELP*-program $\mathcal{P}_8 = (\{\sim x \leftarrow w, w \leftarrow z, z, y\}, \{\sim a \prec\!\!\!- y\})$. When joining $\mathcal{A}$ with $\mathcal{P}_8$, we have that there are strict derivations for both x (from the strict part of $\mathcal{A}$) and $\sim x$ (from the strict part of $\mathcal{P}_8$). Therefore, we apply the function δ to at least one strict rule from $\mathcal{P}_8$ that is involved in the derivation of $\sim x$; for instance, $\delta(w \leftarrow z) = w \prec\!\!\!- z$. Now we have:*

$$\mathcal{P}_8 \circledast^{\omega}_{\mathcal{P}} \mathcal{A} = (\{\sim x \leftarrow w, z, y, x \leftarrow t, t\}, \{\sim a \prec\!\!\!- y, w \prec\!\!\!- z\}) *^{\omega}_{\mathcal{P}} \langle \{a \prec\!\!\!- x\}, a \rangle$$

From this DELP*-program we can build the following extended arguments and dialectical tree:*

$\langle \mathcal{A}, a \rangle : \langle (\{x \leftarrow t, t\}, \{a \prec\!\!\!- x\}), a \rangle$
$\langle \mathcal{B}, \sim a \rangle : \langle (\{y\}, \{\sim a \prec\!\!\!- y\}), \sim a \rangle$
$\langle \mathcal{C}, \sim x \rangle : \langle (\{\sim x \leftarrow w, z\}, \{w \prec\!\!\!- z\}), \sim x \rangle$



Finally, as explained in the previous section, incisions over arguments $\mathcal{B}$ and $\mathcal{C}$ have to be made in order to turn $\mathcal{A}$ into an undefeated argument.

Remark 3 *Revising a program $\mathcal{P}$ by an extended argument $\langle (\Pi', \emptyset), \alpha \rangle$ is the case of an argument that, once introduced into $\mathcal{P}$, would have no argument against it (i.e., by definition, there would be no arguments for $\sim\alpha$), since α would have a strict derivation from Π'. Therefore, strict derivations for $\sim\alpha$ are to be weakened into defeasible derivations that, although they have no effect as arguments for $\sim\alpha$, they can be a part of other derivations that should not be "broken". This stresses the importance of not deleting conflicting strict rules, which can have undesirable collateral effects.*

5 Discussion, Conclusions & Future Work

In this work we have presented two different approaches for the WP Argument Revision operator: one considering regular arguments, and another, considering extended arguments. Both operators have a common theoretical basis, but the latter one has to resolve some extra issues.

The different versions of the *argument revision operator for defeasible logic programs* here proposed are based by most of the previous operator's definition. While *selections* may be related to the **partial-meet contractions** theory [2], *incisions* are inspired by **kernels contractions** [9]. Furthermore, the order established by a preference criterion on selections may be possibly related to **safe contractions** originally exposed in [3], and later on related to kernel contractions in [10]. Indeed,

an argument is a kind of kernel or minimal proof for a given consequence. These concepts are more deeply treated in [4], where the **Kernel Revision by a Set of Sentences** is proposed. Moreover, this operator constitutes part of the inspiration for the **argument revision operators** here exposed.

Our definition of the "$\diamond_{\mathcal{P}}$" operator ensures that there is no inconsistent intermediate epistemic state during the revision process. In [5], a non-prioritized revision operator over explanations "$\diamond_{\mathcal{N}}$" is introduced, which does generate an inconsistent intermediate epistemic state when the revision is performed. The latter operator justifies the non-prioritization by the assertion that there is no reason to accept new information blindly, discarding older beliefs without proper justification. We agree with this posture, but since we are defining an operator that has the objective of warranting the conclusion of the newly inserted external argument, we have to give priority to newer information. Let suppose the definition of a non-prioritized version of our operator by means of an operator "$\diamond_{\mathcal{N}}$":

$$(\Pi, \Delta) \circledast^{\omega}_{\mathcal{P}} \langle(\Pi', \Delta'), \alpha\rangle = (\Pi \diamond_{\mathcal{N}} \Pi', \Delta \cup \delta((\Pi \cup \Pi') \backslash (\Pi \diamond_{\mathcal{N}} \Pi')) *^{\omega}_{\mathcal{P}} \langle\Delta', \alpha\rangle$$

Note that the matter of de-prioritizing the incorporation of new information seems to be attained just to the join of the strict rules sets, while when we refer it as "warrant-prioritized", a reference to the priority of giving warrant to the new conclusion α is given. However, the definition of the non-prioritized operator has a major flaw: it does not ensure the warrant of α. An example will clarify this assertion:

Example 9 *Consider the extended argument* $\mathcal{A} = \langle(\Pi'_9, \{\}), a\rangle = \langle(\{a \leftarrow b, b\}, \{\}), a\rangle$ *and the* DELP*-program* $\mathcal{P}_9 = (\Pi_9, \{\}) = (\{\sim a \leftarrow c, c\}, \{\})$. *Here, if we prefer some older information over new one, we will have no argument for "*a*" from* $(\Pi_9 \diamond_{\mathcal{N}} \Pi'_9, \delta((\Pi_9 \cup \Pi'_9) \backslash (\Pi_9 \diamond_{\mathcal{N}} \Pi'_9)))$*, no matter if the rule* $(a \leftarrow b)$ *is turned into a defeasible rule (see Remark 3). Not having an argument for "*a*" makes impossible to have a warrant for it.*

In general, the theory we are defining cannot be trivially related to the basic concepts of belief revision. Regarding the basic postulates for a revision operator, as originally exposed in [2], a deep analysis is required. For example, the *success* postulate ($K * \alpha \vdash \alpha$) makes reference to a knowledge base K, which in our case is a DELP-program $\mathcal{P} = (\Pi, \Delta)$. For both of the argument revision operators here proposed, success is defined analogously, where the consequence notion is the warrant of the conclusion of the argument being added to $\mathcal{P}$. This statement is verified by Theorem 1. Another interesting postulate to be analyzed is *consistency*, which states that the outcome of a revision $K * \alpha$ must be consistent if α is non-contradictory. In our proposal, this postulate is treated in a trivial manner, since programs are revised by arguments, which are consistent by definition. Regarding extended arguments, a join between the strict parts of the program and the argument is performed to ensure consistency. DELP-programs are divided in two subsets of rules: Π and Δ, where only Π is required to be consistent, which is not modified by any of the argument revision operators we propose (because arguments do not introduce strict rules). Finally, the consistency postulate always hold for the two warrant-prioritized argument revision operators.

Besides exposing a complete list of the basic postulates and the respective axiomatic representations for each argument revision operator, future work also includes the definition of contraction/expansion of a DELP-program by an argument, and a detailed study towards the possibility of duality between the operators of contraction/expansion and revision in argumentation systems.

Optimality is not a property pursued in this work, since there are some cases in which incisions made in a lower level might compromise a smaller amount of program rules. Our major concern here is to define revision operators which just follow correctness regarding the objectives exposed above. That is, the main objective of this article is to present a first approach for revising defeasible

logic programs by an argument. Other possibilities besides warrant-prioritized argument revision are left unaddressed in this paper, as well as variations of the operators here defined. Some of these options are interesting, whereas others are trivial; however, the whole range of possibilities cannot be accounted on a single article. For instance, regarding the minimal change principle, at least two options arise: (1) we might introduce the notion of epistemic importance in an argument level, and thus would incise first those arguments that are less important wrt. the total epistemic order among them; (2) provided that D-trees are an important tool to understand the interrelation among arguments and their influence to the final answer, we might want to preserve the structure of the hypothetical D-tree; hence, incisions would be performed in its lowest levels.

References

[1] DeLP web page: http://lidia.cs.uns.edu.ar/delp.

[2] C. Alchourrón, P. Gärdenfors, and D. Makinson. *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. The Journal of Symbolic Logic*, 50:510–530, 1985.

[3] C. Alchourrón and D. Makinson. On the logic of theory change: Safe contraction. *Studia Logica*, (44):405–422, 1985.

[4] M. Falappa. Teoría de Cambio de Creencias y sus Aplicaciones sobre Estados de Conocimiento. *Ph. D. Thesis*, June 1999.

[5] M. Falappa, G. Kern-Isberner, and G. R. Simari. Explanations, Belief Revision and Defeasible Reasoning. *Artificial Intelligence Journal*, 141(1-2):1–28, 2002.

[6] A. J. García and G. R. Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.

[7] P. Gärdenfors. An Epistemic Approach to Conditionals. *American Philosophical Quarterly*, 18(3):203–211, 1981.

[8] P. Gärdenfors. Knowledge in Flux: Modelling the Dynamics of Epistemic States. *The MIT Press, Bradford Books, Cambridge, Massachusetts*, 1988.

[9] S. O. Hansson. Kernel Contraction. *The Journal of Symbolic Logic*, 59:845–859, 1994.

[10] S. O. Hansson. *A Textbook of Belief Dynamics: Theory Change and Database Updating. Springer.* 1999.

[11] I. Levi. Subjunctives, Dispositions, and Chances. *Synthèse*, 34:423–455, 1977.

[12] V. Lifschitz. Foundations of Logic Programs. In G. Brewka, editor, *Principles of Knowledge Representation*, pages 69–128. CSLI Pub., 1996.

[13] M. Moguillansky and M. Falappa. A Non-monotonic Description Logics Model for Merging Terminologies. *Revista Iberoamericana de Inteligencia Artificial (AEPIA), ISSN 1137-3601*, 2007. at press.

[14] N. D. Rotstein, A. J. García, and G. R. Simari. Reasoning From Desires to Intentions: A Dialectical Framework. *22nd. AAAI Conference on Artificial Intelligence (AAAI 2007), Vancouver, British Columbia, Canadá*, 2007. To appear.

A Tourism Recommender Agent: From theory to practice.

Ana Casali and Armando Von Furth
Depto. de Sistemas e Informática FCEIA - UNR
Av Pellegrini 250, 2000 Rosario, Argentina.
acasali@fceia.unr.edu.ar

and

Lluís Godo and Carles Sierra
Institut d'Investigació en Intel·ligència Artificial (IIIA) - CSIC
Campus UAB, 08193 Bellaterra, Catalunya, España.
{godo, sierra}@iiia.csic.es

Abstract

In this paper a multiagent Tourism Recommender System is presented. This system has a multiagent architecture and one of its main agents, The Travel Assistant Agent (T-Agent), is modelled as a graded BDI agent. The graded BDI agent model allows to specify an agent's architecture able to deal with the environment uncertainty and with graded mental attitudes. We focus on the implementational aspects of the multiagent system and specially on the T-Agent development, going from the theoric agent model to the concrete agent implementation.

Keywords: Recommender Systems, Graded BDI Agents, Tourism, Prolog.

1. INTRODUCTION

In the last years the Artificial Intelligence (AI) community has carried out a great deal of work on recommender systems [12]. This kind of systems can help people to find out what they want, especially on the Internet. Agent technology becomes invaluable by appreciating the facts that we expect these systems to take personal preferences into account, and to infer and intelligently aggregate opinions and relationships from heterogeneous sources and data. Furthermore, we want the systems to be scalable, open, privacy-protecting and we want to get the recommendations with the least possible work on users' behalf [7]. From the application of this technology results a community of distributed, complex and autonomous recommender agents.

Among recommender systems we particularly concentrate on the tourism domain. The travel and tourism industry is one of the most important and dynamic sectors in Business-to-consumer (B2C) e-Commerce. In this context, recommender applications can be valuable tools supporting, for example, information search, decision making, and package assembly. Moreover this is an interesting domain, where diverse user's preferences and restrictions can be considered. Because of this variety, the recommendation systems can be treated in different levels of complexity and the knowledge-based approaches are very suitable [1].

Also, several architectures have been proposed to give agents a formal support. Among them, a well-known intentional formal approach is the BDI architecture proposed by Rao and Georgeff [9]. This model is based on the explicit representation of the agent's beliefs (B), its desires (D), and its

intentions (I). Indeed, this architecture has evolved over time and it has been applied, to some extent, in several of the most significant multiagent applications developed up to now.

We consider that making the BDI architecture more flexible, will allow us to design and develop agents potentially capable of having a better performance in uncertain and dynamic environments. Along this research line we have proposed a general model for Graded BDI Agents (see [2]), specifying an architecture able to deal with the environment uncertainty and with graded mental attitudes. In this agent model, belief degrees represent to what extent the agent believes a formula is true. Degrees of positive or negative desires enable the agent to set different levels of preference or rejection respectively. Intention degrees give also a preference measure but, in this case, modelling the cost/benefit trade off of reaching an agent's goal. Then, agents having different kinds of behavior can be modelled on the basis of the representation and interaction of these three attitudes.

In this work we present the development of a tourist recommender as a case study. The system goal is to recommend the best tourist packages on argentinian destination according to user's preferences and restrictions. The packages are provided by different tourist operators. This system is designed using a multiagent architecture and we particularly use the g-BDI model to specify one of its agents, the Travel Assistant Agent (T-Agent). The purpose of this prototype implementation is to show that the g-BDI agent model is useful to develop concrete agents on real domain.

In previous works we have presented the modelling process of a Travel Recommender Agent using the g-BDI architecture [3] and a general methodology for engineering g-BDI agents [**?**]. In this paper we describe the most relevant aspects of the tourism recommender system implementation and particularly we focus on the T-Agent implementation. This paper is structured as follows, in Section 2 we briefly introduce the g-BDI agent model. Then, in Section 3 the multiagent Tourism Recommender System is presented and in the next Section 4, the principal aspects of the T-Agent implementation are described. Finally, in Section 5 some conclusions are exposed.

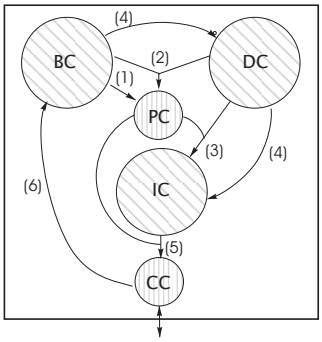
2. GRADED BDI AGENT MODEL

The graded BDI model of agent (g-BDI) allows to specify agent architectures able to deal with the environment uncertainty and with graded mental attitudes. In this sense, belief degrees represent to what extent the agent believes a formula is true. Degrees of positive or negative desire allow the agent to set different levels of preference or rejection respectively. Intention degrees give also a preference measure but, in this case, modelling the cost/benefit trade off of reaching an agent's goal. Thus, a higher intention degree towards a goal means that the benefit of reaching it is high, or the cost is low. Then, Agents having different kinds of behavior can be modeled on the basis of the representation and interaction of these three attitudes.

The specification of the g-BDI agent model is based on Multi-context systems (MCS) [**?**] to allow different formal (logic) components to be defined and interrelated. The MCS specification contains two basic components: units or contexts and bridge rules, which channel the propagation of consequences among theories. Thus, a MCS is defined as a group of interconnected units: $\left\langle \{C_i\}_{i\in I}, \Delta_{br} \right\rangle$, where each context $C_i \in \{C_i\}_{i\in I}$ is the tuple $C_i = \langle L_i, A_i, \Delta_i \rangle$ where L_i, A_i and Δ_i are the language, axioms, and inference rules respectively. When a theory $T_i \subseteq L_i$ is associated with each unit, the specification of a particular MCS is complete. Δ_{br} can be understood as rules of inference with premises and conclusions in different contexts.

The deduction mechanism of these systems is based on two kinds of inference rules, internal rules Δ_i, and bridge rules Δ_{br}, which allow to embed formulae into a context whenever the conditions of the bridge rule are satisfied.

In the g-BDI agent model, we have *mental* contexts to represent beliefs (BC), desires (DC) and



Bridge rule (3)

$$\frac{DC : (D^{+}\varphi, d), PC : plan(\varphi, \alpha, P, A, c, r)}{IC : (I\varphi, f(d, c, r))}$$

Figura 1: Multi-context model of a graded BDI agent and a bridge rule example.

intentions (IC). We also consider two *functional* contexts: for Planning (PC) and Communication (CC). Thus, the g-BDI agent model is defined as the MCS: $A_g = (\{BC, DC, IC, PC, CC\}, \Delta_{br})$.

The overall behavior of the system will depend of the logic representation of each intentional notion in the different contexts and the bridge rules. The specification of the g-BDI agent model, with the logic schema for each context (i.e. the language, axioms and inference rules and a set of basic bridge rules can be seen in [2]. The left side of Figure 1 illustrates the g-BDI agent model proposed with the different contexts and the bridge rules relating them.

The intention degree trades off the benefit and the cost of reaching a goal, by a plan execution. One of the bridge rules included in the agent model (see Figure 1 right side) infers the degree of intention towards a goal φ ($I\varphi$) for each plan α that allows to achieve the goal. This value is deduced from the degree of desire $D^{+}\varphi$ (d), the expected satisfaction of the desire through the plan execution (r) and the cost (c) of the plan. This degree is calculated by a suitable function f.

In order to represent and reason about graded notions of beliefs, desires and intentions, we use a modal many-valued approach where uncertainty reasoning is dealt with by defining suitable modal theories over suitable many-valued logics. For instance, let us consider a Belief context where belief degrees are to be modeled as probabilities. Then, for each classical formula φ, we consider a modal formula $B\varphi$ which is interpreted as "φ is probable". This modal formula $B\varphi$ is then a *fuzzy* formula which may be more or less true, depending on the probability of φ. In particular, we can take as truth-value of $B\varphi$ precisely the probability of φ. Moreover, using a many-valued logic, we can express the governing axioms of probability theory as logical axioms involving modal formulae. Then, the many-valued logic machinery can be used to reason about the modal formulae $B\varphi$, which faithfully respect the uncertainty model chosen to represent the degrees of belief. In this proposal, for the mental contexts we choose the infinite-valued Łukasiewicz logic but another selection of many-valued logics may be done for each unit, according to the measure modeled in each case.

To set up an adequate axiomatization for our belief context logic we need to combine axioms for the crisp formulae, axioms of Łukasiewicz logic for modal formulae, and additional axioms for B-modal formulae according to the probabilistic semantics of the B operator. The same many-valued logic approach is used to represent and reason under graded attitudes in the other mental contexts. The formalization of the adequate logics –language, semantics, axiomatization and rules – for the different contexts is described in [2].

3. TOURISM RECOMMENDER SYSTEM

In this section we present the general architecture of the Tourism Recommender System. The methodological aspects of the analysis and design stages of this case study can be seen in [4].

Inspired in the different members of a Tourism Chain, in the analysis phase we have detected the following roles: the Provider role (tourist package providers), the Travel Assistant role and Services role (hotel chains, airlines, etc.). In this case study we don't deal with the Service role, we only mention it as a necessary collaborator of the Provider role. Other functional roles were captured i.e., the Interface role, to manage the user interface and the Reservory-Maintenance role (R-Maintenance), to charge, translate to an adequate format and discharge the tourist packages that are sent by the Provider role. In this simplified version of Recommender System, we define two agent's types: the Provider agent and the Travel Assistant Agent. We assign the Interface role, the Reservory Maintenance role and the Travel Assistant role to the Travel Assistant Agent (T-Agent). As it is natural in the Tourism Chain, different Tourist Operators may collaborate in the Provider role. To represent these different sources of tourist packages, we use two different agents (P-Agents). This multiagent system is easily scalable to include other providers.

The agents in the Recommender system with the principal source of information they interact with (i.e., the destination ontology and the package reservory), are illustrated in Figure 2.

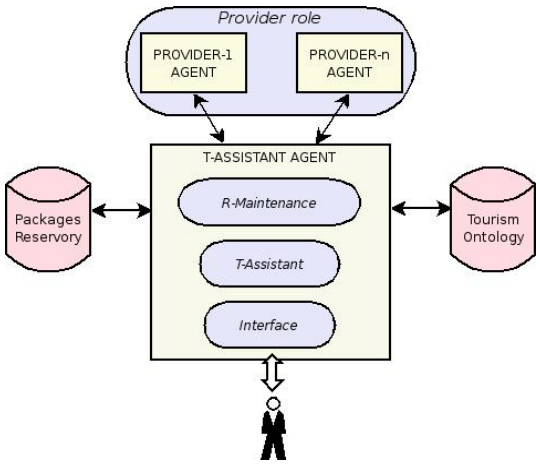


Figura 2: Multiagent architecture for the Tourism Recommender System

The implementation of the Recommender system was developed using SWI-Prolog [11]. This is a multi-threaded version of prolog allowing an independent execution of different contexts (i.e. in different threads). A prior implementation of multi-context agents using this software [6] was a starting point for our development. Furthermore, this prolog version is open source, it is well documented and includes a graphic interface tool in native language.

For our recommender system, each provider agent in the multi-agent systems may be executed in one thread and different threads correspond to the T-Agent components. The software has a set of instructions to deal efficiently with the message communication between threads.

3.1. Providers agents

In our multiagent recommender system two Tour Operator agents (*P-Agents*) are implemented, but the architecture enables to easily include other providers. These agents are only considered as packages suppliers and therefore, we do not get into the inner architecture of them. Each *P-Agent* runs in a different thread being in this way independent from each other and from the *T-Agent*. When the *T-Agent* requests for information, the *P-Agents* send all the current packages they can offer. The communication between agents is by message interchange.

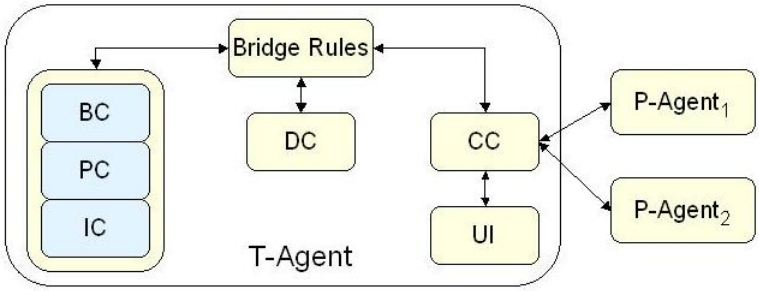


Figura 3: Multithread system scheme

In the real world each tourism operator may structure the tourist packages in a different way and using its own terminology. To experiment with heterogeneous providers, we use different field names in the plan structure used in each *P-Agent*. Then, these structures are translated into the format the *T-Agent* use. Thus, a wrapper functionality is needed and it is carried out by the Communication context of the *T-Agent*. In a more complete multiagent recommender architecture a wrapper agent may be included.

4. T-AGENT IMPLEMENTATION

The principal role of the T-Agent is to give tourists recommendation about argentinian packages. This agent may be suitable modelled as an intentional agent and particularly, by a g-BDI agent model. This agent model is specified by a multicontext architecture having mental and functional contexts (i.e, *BC, DC, IC, PC and CC*) and a set of bridge rules (*BR*s).

Then, for developing the T-Agent the implementation of these interconnected components is needed. Each context has its own inference rules and facts, and they should not interfere. Choosing to use a thread for each context allows the desired separation but could slows considerably the application. The solution adopted for our implementation was to place some of these components in different threads. That is the case for the Communication context (CC), the Desire context (DC) and some bridge rules. Since the Belief (BC), Planner (PC) and Intention (IC) contexts interchange quite a lot of information, for efficiency reasons they run in the same thread. The multithread scheme for the T-Agent in the multiagent system is illustrated in Figure 3, where the yellow boxes represent different threads and the arrows, their interactions.

For this multithreaded implementation the policy adopted following [6], is to have asynchronous threads and asynchronous communication. It means that the messages are sent and received at any time, but they are processed only when the unit is inactive (has finished the internal deductions). Each Unit has got a message queue that retains the messages until they are been processed. A communication meta-interpreter, is devoted to synchronise the ongoing inference process and the arrival of new incoming messages.

In our prototype the principal interchange of messages is during the initial stage. In this phase the *T-Agent* asks the *P-Agents* for the current tourist packages. Answering this request, the *P-Agents* send many messages, each one containing a package offered. The software tool successfully support this intensive messages interchange.

The Communication context (CC) in the *T-Agent* is in charge of receiving these messages, translating them and immediately it sends them to the Belief context (BC). In this way the agent knowledge is increased with the package information. In the next subsections we described how the principal multi-context components of the *T-Agent* are implemented in order to obtain the desired behavior.

We begin with the Communication context that provides the agent with a unique and well-defined interface with the environment.

4.1. Communication Context

The Communication context (CC) constitutes the *T-Agent* interface with its environment and makes it possible to encapsulate the agent's internal structure. This context will take care of the sending and receiving of messages to and from other agents in the multiagent society where our graded BDI agents live. The CC is in charge of interacting with the tourism operators (*P-Agents*) and with the tourist that is looking for recommendation.

In an extended version of this system, all the interactions could be improved in many different aspects (e.g. being more dynamic).

4.1.1. Interaction with the P-Agents

The *T-Agent* before beginning its recommendation task, updated its information about current packages (carrying out its reservory maintenance role). This is achieved by the CC through the following steps:

Require the packages offered - The CC sends a message to each P-Agent asking them for the current touristic packages they offer.

Receive packages and translate them - The CC behaves as a wrapper, translating the incoming packages into the T-Agent format.

Send packages - Once the packages are put into the correct format, they are sent to the Planner context. The recommendation will derive from this package information and domain knowledge.

4.1.2. User interface

The user interface is in charge of explicitly acquiring the tourist's profile, giving him the resulting recommendation and receiving the user's feedback. In a first approach this interface was developed using the native graphic library of the software tool. As a tool requirement, this graphic interface runs in an independent thread in closed interaction with the CC one. This interface may be divided into the next sequential stages:

User's preferences acquisition: These preferences are explicitly acquired asking him to fill in a form. The tourist can select his preferences (positive desires) and restrictions (negative desires) assigning them a natural number from 1 to 10 to represent the level of preference (resp. restriction) in the selected item. Furthermore, he can choose different parameters as: the flexibility of restrictions (flexible or strict), the espected frequency of the selected activity (high or low) and the priority criterion to order the recommended packages (preference satisfaction, minimum cost or trust). An example of a tourist's selection using this interface is shown in Figure 4.

Once the user finishes his selection, the CC sends all the acquired information to the Desire context (DC).

Bring the resulting recomendation: As result of the *T-Agent* deliberation process, the CC receives from the Intention context (IC) a ranking of feasible packages that satisfies some of the tourist preferences. The ranking is ordered also taking into account the priority criterion he has selected

Figura 4: User interface: tourist's preferences (left) and package recommendation (right).

(e.g. preference satisfaction). The first packages of this ranking are showed to the tourist and the user can visualize the information about them opening *pdf* files (may be used other multimedia files).

Receive Tourist's feedback: After analyzing the ranking of the recommended packages the user can express through the interface his opinion about the recommendation. For this task, the options considered are the following:

- Correct: The user is satisfied with the ranking obtained.
- Different order: The given recommended packages are well considered by the user, but are in a different order than the user's own ranking. Then, he is able to introduce the three best packages in its right order.
- Fair: The user is not satisfied with the given recommendation. Then, the interface enables him to introduce a comment about his opinion.

All the information resulting from the previous stages (i.e., the tourist's preferences, the recommendation given and the user's feedback) is stored as to evaluate the system performance.

4.2. Desire Context

As the *T-Agent* is a kind of *personal agent*, its overall desire is to maximize the satisfaction of tourist's preferences. Then, in this context the different tourist's graded preferences and restrictions are respectively represented as positive and negative desires.

On the one hand, the negative desires are used as strong constraints namely, the *T-Agent* will look for packages that will not make true any of them. On the other hand, from the elementary positive desires all their conjunctions are built as combined desires. The *T-Agent* will use all these desires as pro-active elements, looking for different packages that will allow tourists to satisfy any of them.

Then, the theory in this context is constituted by positive and negative desires (represented by $desU$ formulae).

The user's preferences are acquired in the CC by the user interface and are introduced in a list to the DC. The message is first pre-processed by the meta-interpreter. In the following items we describe how the positive desires are built, in a similar way the negative ones are treated:

1. **Elementary desires:** The DC takes each desire from the list received from the CC, normalizes its degree (i.e. mapping it from $\{1, ..., 10\}$ into $(0, 1]$) and adds it to the context formulae. The structure of these formulae is: $desU(y(Desire, Value), NormalizedDegree)$

   The relation $y(Desire, Value)$ represents a positive desire where the first argument is the class of desire (e.g. transport "transporte") and the second is the value the tourist has chosen (e.g. plane "avion"), followed by the normalized degree (e.g. 0.8). For instance, elementary desires in DC for a tourist's consultation (see selection in figure 4) are:

   ```
   desU(yLst([(zona, patagonia)]), 0.9)
   desU(yLst([(transporte, avion)]), 0.7)
   desU(yLst([(comodidad, hotel3)]), 0.5)
   ```

2. **Combined Desires:** After the elementary desires are added to the context, all the possible conjunctions are built. The degrees for the conjunctions are calculated following the guaranteed possibility model (see [2]) where the resulting degree is greater than the maximum of elementary degrees. Having in the DC formulae like $desU(y(D_1), G_1)$ and $desU(y(D_2), G_2)$ it is added the combined desire $desU(yLst([D_1, D_2]), G)$. In this prototype G is particularly computed by the following function:

   ```
   calcularGraduacion(G1, G2, G) :-  G is G1 + ((1 - G1) * G2)
   ```

   As for example, it is showed the code of one of the conjunctive combinations built from the elementary desires given above:

   ```
   desU(yLst([(zona, patagonia), (transporte, avion)]), 0.92)
   ```

We notice, that in the special case of some desire types (related to accommodation and particular resources) we consider that the tourist will also be satisfied (in some degree) if he receives from the package a similar value than the desired one. In these cases we are considering implicit rules like "If the tourist have a positive desires D in degree d and D is similar to D' in a degree s, then he also desires D' in a degree $d' = f(d, s)$". Even this rules are not explicitly coded in DC, they are used for the expected satisfaction computation (see next subsection 4.3.4).

The positive and negative desires are both passed by a bridge rule to the Planner context where the feasible packages to satisfy the tourist are selected.

4.3. Belief Context

In this context the *T-Agent* represents all the necessary knowledge about tourism and the argentinian domain: tourist packages, information about destinations and rules to infer how the different user's preferences may be satisfied (in some degree) by the feasible tourist packages.

4.3.1. Tourist packages

One of the most significant data structure in our system is the package structure. After analyzing nearly fourty argentinian packages selected from the Internet, a general structure capable of representing the information available in most of them, was proposed. Each package is represented as a list containing an identifier, a tour provider, the package cost and a travel-stay sequence represented by $Trip$ as can be seen in the following structure:

Package	::=	(Id, Provider, Cost, Trip)
Trip	::=	[(Travel, Stay)]
Travel	::=	(Transport, Road)
Stay	::=	(Destination, Days, Accommodation, [Activity])
Activity	::=	activity(Sport, Hours) | excursion(Resource, Hours, Name)

As for example, the prolog representation of the package named *holCalafatePatagonia* is presented below:

```
paq(id(holCalafatePatagonia), costo(1900),
  [(viaje(avion, aire), estadia(calafate, dias(3), comodidad(apart),
    actividades([
      [act(cityTour), horas(4)],
      [exc(parqueNacional), horas(8), peritoMoreno]]))),
  (viaje(avion, aire), estadia(ushuaia, dias(4), comodidad(hotel3),
    actividades([
      [act(cityTour), horas(1.5)],
      [exc(museo), horas(1), finDelMundo],
      [exc(historia), horas(1), carcelDeReincidentes],
      [exc(parqueNacional), horas(2), tierraDelFuego],
      [exc(lago), horas(1), escondido],
      [exc(lago), horas(1), fagnano]]))),
(viaje(avion, aire), null)])
```

Notice that in the last element of the travel list, the stay is null representing the return travel.

4.3.2. Destination ontology

The *T-Agent* needs to have information about the country and the different possibilities its diverse places bring. Usually the packages have little information about the destinations and the resources available in them. This domain knowledge is complementary to the package information and vital to infer how a trip including certain destinations, can satisfy some tourist preferences (e.g. natural resources). To structure the knowledge about argentinian tourism, we analyzed different tourism ontologies and most of them were focussed on destinations (see e.g. [8]) including the resources they have, the activities they offer, etc. Inspired in some of them, the following features were extracted for the destination ontology in our prototype.

Destination	::=	*(Name, Coordinates, Zone, [NaturalResource], [ArtificialResource], [Activity])*
Coordinates	::=	*(X, Y)*
RecursoNatural	::=	*Resource*
RecursoArtificial	::=	*Resource*
Resource	::=	*(KindOfResource, Name)*

The information of almost fifty argentinian destinations (i.e. all the places related to the packages used) was introduced to fill in this ontology. This information was extracted from official web-sites.

We use as *coordinates* the geographic coordinates provided by tInstituto Geográfico Militar de la República Argentina (http://www.geoargentina.com.ar). The *zone* assigned to each destination corresponds to the partition of argentinian provinces into zones, proposed by Secretaría de Turismo de la República Argentina (http://www.turismo.gov.ar).

An example of the destination structure for the *Ushuaia* city is presented below:

```
localidad(nombre(ushuaia), provincia(tierraDelFuego),
   gps(54.80, 68.31), zona(patagonia),
   naturaleza([(parqueNacional,tierraDelFuego), (canal,beagle),
```

```
    (bahia,lapatala), (lago,roca), (lago,fagnano),
    (lago,elEscondido), (laguna,negra), (rio,grande)]),
infraestructura([(museo,finDelMundo), (museo,regional).
    (museo,acatushun), (historia,presidio),
    (ingenieria,trenFinDelMundo)],
actividades([avistajeFauna,esqui,navegacion,pesca,trekking]))
```

The ontology used in this prototype was directly code in a prolog file, but it is possible for the *T-Agent* to receive an ontology built using an ontology editor (via XML code).

4.3.3. Special Relations in the domain

To increase the domain knowledge of the *T-Agent* other relations were included in the BC.This knowledge about related concepts makes it possible for the *T-Agent* to expand the search to other terms related to the ones expressed in the tourist's preferences and are used in the selection of the best packages for the tourist. In this implementation we considered important to include the following relations:

1. **Similarity dictionary:** The BC includes a set of similarity relations between synonymous or similar concepts, according to the tourism domain. As the *T-Agent* has a Belief context that deals with graded information, this similarity relation may include a degree $g \in [0, 1]$ expressing the semantical distance between terms. The formulae in this dictionary are structured as: $belU(similar(term_1, term_2), g)$

   For instance, we show a fragment of this silimarity dictionary:

   ```
   % accommodation category
   belU(similar(apart, hotel3), 0.75)
   belU(similar(camping, campamento), 1.0)
   % nature category
   belU(similar(lago, embalse), 0.7)
   belU(similar(montaña, serro), 0.8)
   ```

2. ***"Better than"* relation:** For the accommodation concepts was added a "better than" relation expressing whether an accommodation is better than another one. This transitive relation allows the *T-Agent* to expand the search of the packages that satisfy the user's preferences, to those that include accommodations better than the selected one.

4.3.4. Preference Satisfaction Estimation

The Belief context is in charge of evaluating the estimation of how a tourist's desire D is satisfied after executing certain package α_P. From the belief degree r of $B([\alpha_P]D)$ and the degree d of desire D, this estimation is computed as the product $E = r.d$. Following the model presented in [2], the truth degree of $B([\alpha_P]D)$ is considered the probability of having *D* after following plan α_P. Since the degree of desire d is provided in the DC formulae, to complete the satisfaction estimation we need rules in the BC to compute the belief degree r of the formula $B([\alpha_P]D)$ according to the different desire types.

Basically a tourist plan may be considered as a time sequence of subplans (see [3] for details), and the satisfaction estimation depends on how much a preference is expected to be satisfied in each stage of a trip. As it was presented above, the packages are structured as: $Package ::= (Id, Provider, Cost, Trip)$ where $Trip$ is a travel-stay sequence $[(Travel_i, Stay_i)]\ i = 1, n$. In our approach each $Travel_i$ and $Stay_i$ parts of the $Trip$ are considered as atomic package stages (sub-plans), amenable to satisfy desires. Packages α_P are therefore modelled as composed plans, $\alpha_P = \alpha_1; ...; \alpha_n$, alternating travel and stay sub-plans.

Then, the belief degree r of $B([\alpha_P]D)$ will depend on the probabilities r_i of having D after the execution of the each sub-plan α_i. For computing the degree r the T-Agent needs to estimate the components r_i and then to aggregate them using a suitable operator, i.e.

$(B([\alpha_1]D), r_1) \wedge ... \wedge (B([\alpha_n]D), r_n) \rightarrow (B([\alpha_P]D), \oplus_{i=1,n} r_i)$, where $\oplus$ is an appropriate aggregation operator.

A set of rules which play this aggregation role, depend on the kind of desire and on the user's priority criterion. In the following items we give the insights of this estimation for positive desires.

- Elementary desires

For evaluating the expected satisfaction of a desire D by executing a package α_P depending on the kind of desire D, the travel or stay stages of α_P are considered. As for example, if the desire is about transport, the travel stages are used, and if it is related to a natural resource, the stay parts are considered.

The underlying idea to compute the expected satisfaction is to take the proportion of the package where the user's desire is expected to be satisfied (in some degree) respect to the total trip. Furthermore, the estimation of how the different stages of a trip may satisfy a preference, may be also graded. In our approach we consider for this estimation, the similarity degree between the tourist's desire and the respective proposal in the package.

On the one hand if the offer is exactly or "better than" the user's preference, the expected satisfaction of achieving the chosen preference in this package stage, is consider the desire degree (i.e. $E = d$). On the other hand if the offer is similar, our approach is to take the similarity degree s (between asked and offered preferences) to compute the expected satisfaction of the user's desire by a package stage (i.e. $E = s.d$).

Then, if the package α_P is composed by different stages, i.e. $\alpha_P = \alpha_1; ...; \alpha_n$ the general way of computing the preference satisfaction of D by the execution of package α_P, i.e the degree r in the formula $(B[\alpha_P]D, r)$, is the next:

$$(B([\alpha_1]D), r_1) \wedge ... \wedge (B([\alpha_n]D), r_n) \rightarrow (B([\alpha_P]D), \frac{\sum_i r_i \times Time_{\alpha_i}}{TotalTime})$$

where $Time_{\alpha_i}$ and $TotalTime$ are computed according to the kind of desire D.

For instance, if D is about acommodation, $Time_{\alpha_i}$ computes the duration (in days) of the stay α_i and $TotalTime$ is the total duration of the trip. In the case of being D an activity and considering that the user has chosen the high activity frecuency, $Time_{\alpha_i}$ is the hours that the activity is programmed in the stage α_i of the package and $TotalTime$ is an estimation of the total number of hours that the activity could take along the trip.

Example: Let us assume a tourist has an accommodation preference of Apart Hotel represented by the desire D: $desU(yLst([(comodidad, apart)]), 0{,}7)$. Then, the *T-Agent* wants to evaluate the expected satisfaction of D through the package *holCalafatePatagonia* (see subsection 4.3.1). This package has two stay stages: in Calafate destination, with accommodation in Apart Hotel and in Ushuaia, providing a Hotel 3*.

Then, the degrees r_i are computed respectively as: $r_1 = 1$ and using the similarity relation $belU(similar(apart, hotel3), 0{,}75)$, $r_2 = 0{,}75$. Finally, the degree r corresponding to the belief degree of $B([holCalafatePatagonia]D)$ is computed as:

$r = \frac{(1 \times 3d) + (0{,}75 \times 4d)}{7d} = 0{,}857$

and the expected satisfaction of the selected preference through this package is $E = r.d = 0{,}599$.

When the tourist's desire is related to a destination resource (e.g. natural resources, activity) the belief degree of achieving the chosen preference by a plan execution, has another interesting characteristic. We have noticed that usually the packages have limited information about destinations and their resources. Thus, for the estimation of some preference satisfacion the *T-Agent* needs to use domain knowledge. In our prototype this information is structured in the destination ontology.

Using the same schema for evaluating the degree r presented above, the computation of r_i degrees are refined. They are computed after a *package-destinations* cross inference, to assess the presence of the tourist's selected preference in the package and in the destination information.

The strategy followed is to evaluate for each package stage α_i, the probability that it has of providing certain resource either it is explicitly offered in a package (r_{Pi}) or it is inferred by the ontology information (r_{Oi}). In this approach the T-Agent takes as the degree r_i the maximum of both estima-

tions, i.e. $r_i = max\{r_{Pi}, r_{Oi}\}$.

- *Combined desires*

The DC theory includes conjunction of positive desires. To evaluate the probability of reaching the conjunction of elementary desires (D_1 and D_2) by the execution of a package α, we assumed that the desires are stochastically independent for each package P. Then, from the degrees r_1 and r_2 corresponding to the elementary desires, we can compute the belief degree in achieving their conjunction by executing the plan α, using the following rule:
$((B[\alpha]D_1, r_1), (B[\alpha]D_2, r_2)) \rightarrow (B[\alpha](D_1 \wedge D_2), r_1 \cdot r_2)$

We remark that the many-valued model of information representation and reasoning in the BC has many advantages. First, this model ables an expressive representation of the domain knowledge. Secondly, this approach allows the agent to evaluate in a more real way the expected satisfaction of preferences by the execution of diverse packages by a *package-ontology* cross inference. Finally, the treatment of this many-valued information makes it possible to compute in a graded way each expected satisfaction, giving the agent more information than a bi-valued approach.

4.4. Planner Context

The PC unit is vital for the *T-Agent* implementation, its theory is composed by $planner$ formulae. This context is responsible for looking for *feasible package*. A *feasible package* satisfies one of the positive desires (elementary or combined) and its execution cannot satisfy any restriction. These feasible plans are computed within this context using an appropriate searching method, that takes into account beliefs and desires injected by bridge rules from the BC and DC units, respectively.

Then, from the positive and negative desires, the package information, the beliefs of the agent about package destinations, the estimation of desire satisfaction by plan execution and the package cost; the Planner can find feasible packages (coded as $paqSi$ formulae). These touristic packages may fulfill the tourist's positive desires, but avoiding negative ones (do not satisfy them in degree greater than a threshold: $UmbralN$).

The following forward rule code this in the Planner context:

```
des(yLst(DeseosP), _), des(nLst(DeseosN), UmbralN),
planner(paq(IdPaq, Proveedor, Costo, _Recorrido)),
bel(contiene(IdPaq, DeseosP), R),
bel(not(contiene(IdPaq, DeseosN)), UmbralN),
bel(costoNormalizado(Costo, CN), 1)
--:  planner(paqSi(IdPaq, Proveedor, CN, DeseosP), R)
```

Notice that this rule uses formula coming from DC (des formulae) and from BC (bel formulae). For each feasible package named $IdPaq$ the normalized cost ($CN \in [0, 1]$) is computed and used instead of its actual cost.

After the PC has found the set of feasible packages, they are passed to the Intention context in charge of making a ranking of these packages.

4.5. Intention Context

The *T-Agent* in this context (IC) finds the intention degree for each feasible package. As we have previously mentioned, the intention degree trades off the benefit and the cost of reaching a goal or desire, through a plan execution.

There is a bridge rule that infers the degree of $I_\alpha(D)$ for each package α that allows to achieve D. This value is deduced from the degree of desire D (GD), the belief degree in achieving D by executing the plan α (GR), the cost of the plan α (CN) and the trust in the package provider (GT).

The intention degree for $I_\alpha(D)$ is calculated by a function f that suitably combines all these factors. Different functions can model different individual agent behaviors. In the *T-Agent* this function is defined as a weighted average:

$$f = (w_d * GD + w_r * GR + w_c * (1 - CN) + w_t * GT)$$

where the different weights w_i are set by the *T-Agent* according to the priority criterion selected by the user (minimum cost, preference satisfaction or trust).

The following bridge rule infers the intention formulae related to the package Id with the corresponding intention degree (G):

```
planner(paqSi(Id, Proveedor, CostoN, DeseosP), GR),
  bel(trust(Proveedor), GT),
  des(yLst(D), GD),
  bel(prioridad(PU), 1),
  f(GD, GR, GT, CN, PU, G)
  --:  int(paqRecomendado(Id), G)
```

Once the rule has been applied to all the feasible plans the IC has a set of graded intention formulae. Using the intention degree the *T-Agent* makes a package ranking that communicates to the CC. We opted to select the first N [1] packages to recommend the tourist.

Finally, the selected packages are passed to the CC and then, through the user interface the *T-Agent* gives the tourist this ranking as recommendation. For instance, in Figure 4 it is showed a tourist's preference selection (on the left) and the resulting recommended ranking (on the right). After analyzing the results, the user is asked to give the system his feedback.

5. CONCLUSIONS

A prototype of multiagent Tourism Recommender system has been implemented. A multiagent approach is suitable for this kind of systems dealing with heterogeneous and distributed information. Particularly we used a g-BDI architecture for modelling the T-Agent, showing in this way, that this model is useful to develop concrete agents in real domains.

We remark that the many-valued model of information representation and reasoning in the g-BDI agent, has many advantages for this implementation. First, this model enables an expressive representation of the domain knowledge (agent beliefs), the user's preferences (desires) and the resulting intentions. Secondly, the implemented approach allows the agent to expand the retrieval of feasible packages using similarity relations and domain knowledge, not explicitly included in the package information. Also, the treatment of many-valued information makes it possible to compute in a graded way the expected satisfaction of the different tourist's preferences, by the execution of diverse packages. Finally, the intention degree of a plan towards a desire satisfaction may be computed as a function of diverse factors (e.g. satisfaction, cost, trust). As we can obtain diverse agent behaviors defining different functions for intention computation, these become a crucial point in the agent model.

The first experimentation of this prototype was carried out with good results. Considering 150 recommendations, the 75 % of the user's opinions were aceptable (correct or different order) and among them, the 70 % was correct. Now we are working in the adjustment of the T-Agent behavior using this user's feedback. As for future work we plan to simulate a crisp BDI version of the T-Agent as to experiment and compare with the graded BDI model of this agent.

[1] N is an agent parameter currently set as $N = 6$

REFERENCIAS

[1] Burke R. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, 69, 2000.

[2] Casali A., Godo Ll. and Sierra C. Graded BDI Models For Agent Architectures. Leite J. and Torroni P. (Eds.) *CLIMA V, LNAI* 3487, pp126-143, Springer-Verlag, Berling Heidelberg, 2005.

[3] Casali A., Godo Ll. and Sierra C. modelling Travel Assistant Agents: a graded BDI Approach. Proceedings of the IFIP-AI, WCC, Volume 217, *Artificial Intelligence in Theory and Practice*, Ed. Max Bramer (ISBN 0-387-34654-6). Boston: Springer, 415-424, 2006.

[4] Casali A., Godo Ll. and Sierra C. A Methodology to Engineering Graded BDI Agents. Proccedings WASI, *CACIC 2006*, 12 pag . Potrero de Funes, Argentina, 2006.

[5] Ghidini C. and Giunchiglia F. Local Model Semantics, or Contextual Reasoning = Locality + Compatibility Artificial Intelligence,127(2):221-259, 2001.

[6] Giovannucci Andrea. Towards Multi-Context Based Agent Implementation, Technical Report IIIA-CSIC, 2004.

[7] Niinivaara, O., Agent-Based Recommender Systems. Technical Report, University of Helsinki, Dept. of CS, 2004.

[8] Prantner Kathrin, E-Tourism, DERI Innsbruck,5 October 2004 (www.deri.org)

[9] Rao A., Georgeff M. BDI Agents from Theory to Practice, Technical Note 56, AAII, April 1995.

[10] Ricci F., Travel recommender Systems, in IEEE Intelligent Systems, November/December 2002, pages 55–57, 2002.

[11] http://www.swi-prolog.org

[12] L. G. Terveen and W. Hill. Beyond Recommender Systems: Helping People Help Each Other. In Carroll, J. (ed.), *HCI in the New Millennium*. Addison Wesley, 2001.

Design and Implementation of a FIPA based Agent Communication Model for a Logic Programming Framework

Mariano Tucat
mt@cs.uns.edu.ar

Alejandro J. García
ajg@cs.uns.edu.ar

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Artificial Intelligence Research and Development Laboratory
Department of Computer Science and Engineering, Universidad Nacional del Sur,

Abstract

In this paper we consider the design of an agent communication model based on the FIPA Architecture and FIPA Interaction Protocols. Our goal is to allow the agents to reach each other by their characteristics and to exchange messages using a standard Agent Communication Language. We propose the design and implementation of a framework as an extension of Prolog, following the spirit of Logic Programming. In our framework, the agents are allowed to register themselves in MASs in order to allow other agents to locate them easily. The agents may search for agents having specific characteristics, or providing determined services in order to interact with them.

Keywords: Agent Communication, Interaction Protocols, Logic Programming

1 INTRODUCTION

Interaction is an essential characteristic of Multi-Agent Systems (MAS). Agent interactions are usually performed by exchanging messages according to some conversation policy or by executing services upon requests made by other agents. Therefore, the ability to locate other agents and communicate with them are features that need to be implemented in agents that are part of a MAS.

The goal of this paper is to design an agent communication model based on the FIPA Architecture [6] and FIPA Interaction Protocols [5]. Thus, our model should allow the agents both to reach each other by their characteristics and to exchange messages. The message exchange should be done using standard Agent Communication Languages (ACLs), in order to allow the interaction among heterogeneous agents.

Suppose as an example that there exist some agents selling books and other agents buying books on behalf of their users. The agents buying books will need to find the agents selling books

Partially supported by CONICET (PIP 5050) and SGCyT Universidad Nacional del Sur (24/ZN11)

and interact with them, but only the ones that sells the books they are interested in. Another similar situation may arise when we have agents controlling the printers of an organization and there exist agents that are trying to print documents on behalf of their users.

The theory of agent communication languages and conversation policies has received a lot of attention in recent years [1, 3, 11, 14]. However, the creation of tools for implementing these formalisms has been progressing at a slower pace, and it is clear that the techniques resulting from such contributions will only be widely adopted when suitable programming languages and tools are available.

There exist different alternatives for implementing MASs. One alternative is to implement the whole MAS ad-hoc. Another way is to use a MultiAgent development framework such as JACK [10], JADEX [13] or 3APL [9]. Finally, another alternative is to implement an MAS using a Programming Language extension that provides the capabilities of finding other agents and exchanging messages.

The alternative of implementing the whole MAS ad-hoc means that the developer of the system is allowed to choose the architecture of each agent, the way they interact and also the way they locate each other. Thus, this alternative has the advantage of a great flexibility in the design and implementation of the system. However, the main disadvantage is that the developer may have to implement everything, including the mechanisms used to locate the agents and also the primitives for exchanging messages.

The alternative of using a MultiAgent development framework such as JACK, JADEX or 3APL, has the advantage of reducing the amount of work needed to implement the system. However, this alternative constrains the developer of the systems to use a specific agent architecture and may also determine the way in which the agents should exchange messages.

Finally, the alternative of implementing the MAS using a Programming Language extension or a framework that provides the capabilities of finding other agents and exchanging messages (such JADE [2] or MadKit [8]) allows a flexible design of the system. This way of implementing the MAS tries to maintain the advantages of the alternatives mentioned before also avoiding their disadvantages.

Our proposal corresponds to the last alternative. Since Logic Programming is widely adopted for the development of intelligent agents, we propose to design and implement a framework as an extension of this language. The extension consist of a set of primitives that follows the spirit of Logic Programming: *to provide a specification of the solution and to hide as much of the implementation details as possible.*

The proposed framework should provide a reliable way for programming communicative agents without dealing with low-level details such as the actual location of an agent. In our framework, agents register themselves in MASs in order to allow other agents to locate them easily. The agents may group themselves by their common characteristics, or by the services provided. Thus, any agent may search for agents having specific characteristics, or providing determined services. Whenever an agent registers itself, it sets the name that other agents will use to identify it.

2 THE PROPOSED AGENT COMMUNICATION MODEL

The main goal of our model is to allow the agents to reach each other by their characteristics or services, without worrying about their locations. Once reached, the agents will exchange messages. This message exchange should be held using standard Agent Communication Languages (ACLs), in order to allow the interaction among heterogeneous agents (*i. e.*, agents developed using different languages or technics).

Thus, in our model, agents register themselves in MASs in order to allow other agents to locate them easily. The agents may group themselves by their common characteristics, such as the type of printer the agent controls, or by the services provided, such as selling science fiction books. Thus, any agent may search for agents having specific characteristics, or providing determined services.

Whenever an agent registers itself, it sets the name that other agents will use to identify it. The agent may choose any desired name, especially a location independent name. Since this name will be used to identify the agent, it must be unique. The entity in charge of maintaining the registered agents is responsible for the uniqueness of the agent names. This entity may reject any agent trying to register itself with an already used name.

Figure 1 depicts our Agent Communication Model. An agent in our model is an standalone program with the capability of interacting with other agents through its Communication Library (CL). The CL provides communication, thus allowing the agent to register itself in different MASs, searching for specific agents and exchanging messages with them. In our model, there exists a special agent, called Yellow Pages Agent (`YPA`), responsible for maintaining all the information of the agents registration in the different MASs and also guarantying the uniqueness of the names identifying the agents.

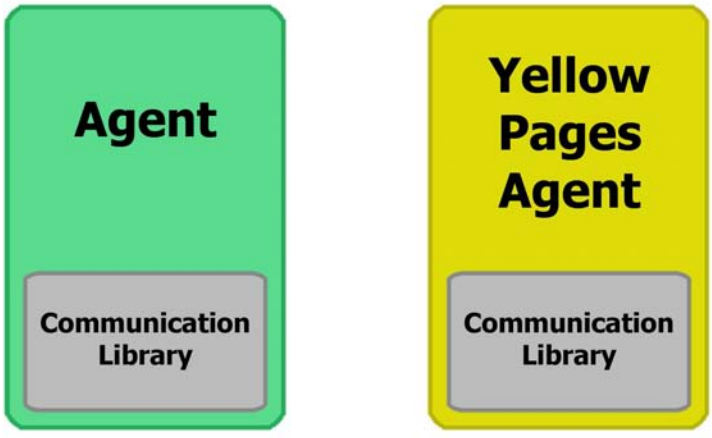


Figure 1: The Proposed Agent Communication Model

Since we based our model in the FIPA Standards [4], it has some similarities with the reference model provided by FIPA, but it also has some differences as we will explain next. The FIPA Reference Model (see Figure 2) includes the agents, the Agent Management System (AMS), the Directory Facilitator (DF) and the Message Transport System (MTS), all of them being part of the Agent Platform (AP), that provides the physical infrastructure in which agents are deployed.

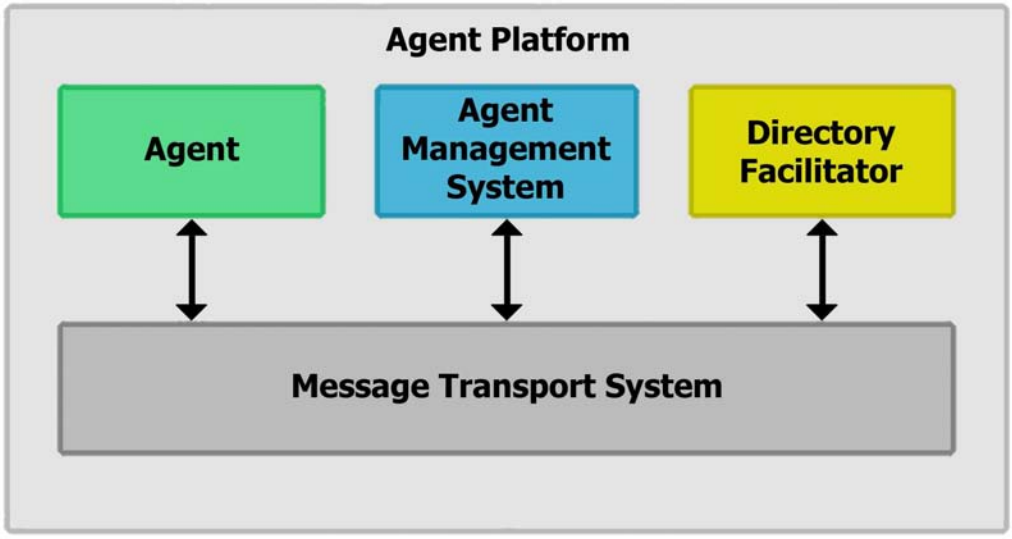


Figure 2: FIPA Reference Model

The AP consists of the machine(s), operating system, agent support software, FIPA agent management components (DF, AMS and MTS) and agents. In our model, we propose that each agent should be an stand-alone program capable of interacting directly with other agents. Thus, we do not provide an AP in which the agent are deployed.

Another difference with the FIPA Architecture corresponds to the existence of the AMS. The AMS exerts supervisory control over access to and use of the AP. The AMS also maintains a directory of AIDs which contain transport addresses for agents registered with the AP. In our model, we avoid the existence of the AMS since the AP does not exist. The directory of AIDs containing transport addresses will be held by the `YPA`, and these addresses will be kept hidden to the agent developer, which will use only the name chosen by the agents.

Our `YPA` is based on the DF provided by the FIPA Reference Model. The DF provides yellow pages services to other agents. Agents may register their services or characteristics with the DF or query the DF to find out what services are offered by other agents. In our case, our `YPA` has also the responsibility of guarantying the uniqueness of the names identifying the agents registered.

As we will see in detail in Section 3, the agents in our model uses FIPA ACL and FIPA Interaction Protocols in order to interact with the `YPA`, thus allowing any agent FIPA complaint to interact with them. In order to interact, the agents have the CL, which is similar to the MTS, with the only difference that our CL is used to exchange messages among any agent, whereas the MTS is the default communication method between agents on different APs.

In order to facilitate the development of agents in our model, we propose a small set of primitives that allow them to interact with the `YPA`, registering themselves and also searching for specific agents. This set of primitives includes two primitives that allow the agent to connect to a specific `YPA` (`connect` and `disconnect`), two primitives for registering specific services or characteristics (`register` and `deregister`) and finally, two primitives for searching for agents with specific characteristics or providing certain services (`which_agents` and `which_MASs`).

- connect(+Name, -Error)

- disconnect(-Error)
- register(+Characteristics, +Ontology, -Error)
- deregister(+Characteristics, +Ontology, -Error)
- which_MASs(-List_Of_MASs, +Ontology, -Error)
- which_agents(-List_Of_Agents, +Characteristics, +Ontology, -Error)

Suppose that we want to develop a system that decides, on behalf of the user, which is the better printer to use in order to print a specific document. Thus, one way to accomplish this task is to develop an agent for each printer and agents that interact with them in order to print documents. In our model, the agents controlling the printers will register themselves in MASs according with their characteristics and the agents willing to print documents will search for agents in specific MASs depending on the desired characteristics of the printer.

The agents controlling the different printers will connect to the `YPA`, using the primitive `connect/2`. Suppose that we have two monochromatic laser printers and that we also have three color ink-jet printers. Thus, the agents controlling these printers will use the primitive `connect(Name, Error)` (see 1 on Figure 3), in order to be able to interact with the `YPA`.

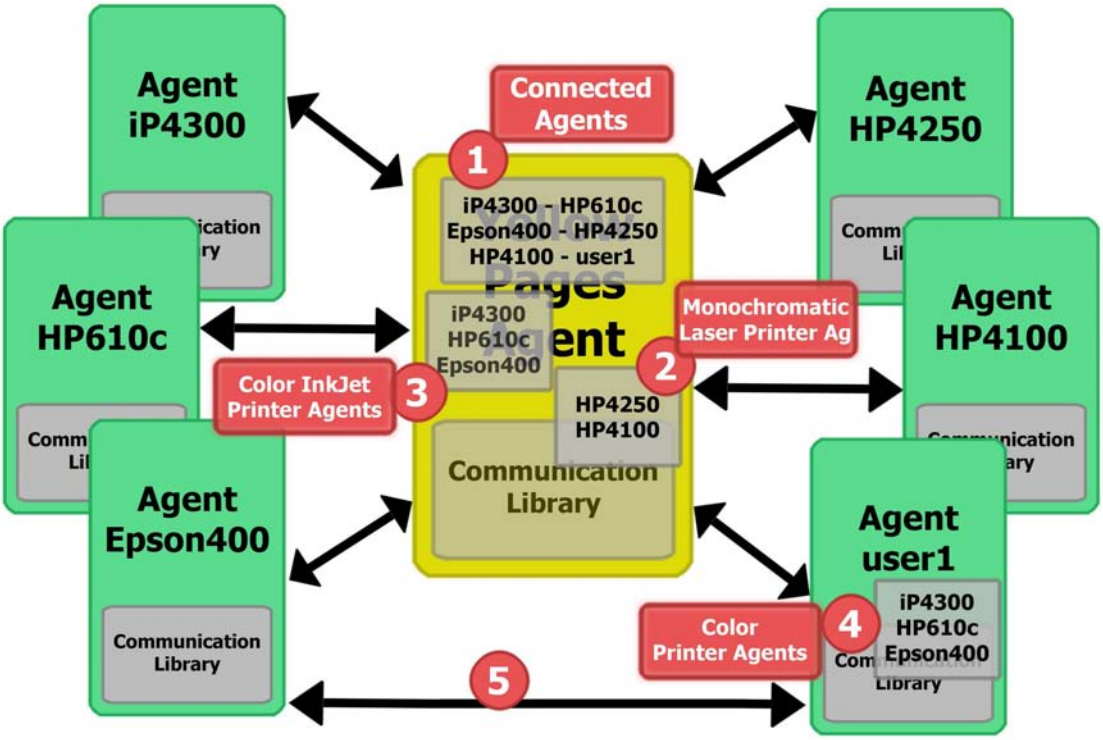


Figure 3: The agents interacting with the `YPA`

Once connected with the `YPA`, they will register themselves, depending on their characteristics, using the primitive `register/3`. For example, an agent controlling a monochromatic laser printer may registry itself calling `register(printers([color(no), type(laser)]), computer_printers, Error)` (see 2 on Figure 3), thus identifying its characteristics and determining the ontology of the terms used. An agent controlling a color ink-jet printer may use `register(printers([color(yes), type(ink-yet)]), computer_printers, Error)` (see 3 on Figure 3).

After that, any agent willing to print a specific document will have to connect to the YPA and then search for the needed printer. In order to search for the existent MASs, the agents may use which_MASs/3. It may call which_MASs(List, computer_printers, Error), obtaining in List the existent MASs (*i. e.*, printers([color(no), type(laser)]) and printers([color(yes), type(inkjet)])).

In the case that the document to print is in color, the agent will try to find a color printer, looking for agents in the MAS having printers with this characteristic. Thus, it may call which_agents(List, printers([color(yes)]), computer_printers, Error). In our example, the agent user1 will obtain in List the names of the three color ink-jet printers (see 4 on Figure 3). Then, the agent may interact with any of the agents controlling the color printers, for example, it may interact with the agent Epson400 (see 5 on Figure 3).

3 FRAMEWORK IMPLEMENTATION

As mentioned above, in order to facilitate the development of agents in our model, we propose a set of primitives that allow the agents to interact with the YPA. This set of primitives is provided by the CL, and in this section we will explain in detail these primitives and one way of implementing them in Prolog. Our framework also provides an implementation of the YPA that supports the registration of agents and the search of them through their characteristics or services provided.

This set of primitives was designed and implemented as an extension of Logic Programming since this language is widely adopted for the development of intelligent agents. These primitives provide a reliable way for programming communicative agents without dealing with low-level details such as the actual location of an agent. The framework also facilitates the use of sophisticated Knowledge Representation and reasoning formalisms already developed for LP.

As we mentioned in section 2, we use FIPA ACL and the FIPA Interaction Protocols in order to interact with the YPA, thus allowing any agent FIPA complaint to interact with our agents. The YPA implements the interaction protocols defined by FIPA, therefore any agent, developed from scratch or using another FIPA complaint framework, interacting with it and following these standard protocols, will obtain the corresponding answers.

The set of primitives proposed allows the agents to register themselves and also to search for specific agents. This set includes two primitives to connect to a specific YPA (connect and disconnect), two primitives for registering specific services or characteristics (register and deregister) and two primitives for searching for agents with specific characteristics (which_agents and which_MASs).

The primitives to connect to a specific YPA have as primary goal to establish the connection between the agent and the specified YPA. Once the connection is established, the agent will register itself with a proposed name and the YPA may agree or refused the request. In the case that the chosen name is unused, the YPA will agree, whereas in the other case, it will refuse.

A simplified version of the predicate connect/2 can be seen in Figure 4. The predicate creates an address to receive messages from other agents (calling listen/1), and then it establishes the connection with the YPA (calling establish_connection/3), using the obtained address (get_ypa_address/2). The address of the YPA can be set using a specific predicate (set_ypa_address/2). Note that it may connect with any YPA running on the same network.

```
connect(+Name, -Error) :-
  listen(-Port),
  get_ypa_address(-Host, -Port),
  establish_connection(+ypa, +Host, +Port),
  send_connection_request(+ypa, +Name),
  get_answer(-Answer),
  ( Answer = done,
    Error = no_error
  ;
    Answer = failure,
    Error = used_name(+Name)
  ).
```

Figure 4: A simplified version of the `connect/2` predicate

After the connection is established, the agent starts the interaction sending the corresponding request message to the `YPA` (`send_connection_request/2`) and waits for the answer. Depending on the result of the interaction, the term `Error` will be instantiated with `no_error`, indicating that the connection was successfully accomplished, or it may be instantiated with `used_name(Name)`, indicating that the name is already used and that the agent should choose another one.

The predicate `disconnect/1` closes the connection established with the `YPA`, also eliminating any registered characteristic. Once done, the agent will not be allowed to register any characteristic or service provided, or even to search for any specific agent, until it connects again with an `YPA`. In order to do this, the predicate will first interact with the `YPA` using the corresponding protocol and then it will close the connection.

The primitive `register` allows the agents to register specific services or characteristics into different MASs, thus, allowing other agents to locate them easily. In the case of the `deregister`, it allows the agents to eliminate any characteristic or service registered before. Both primitives require that the agent and the `YPA` have already established a connection using the primitive `connect/2`, explained above.

In Figure 5, a simplified version of the primitive `register/3` is shown. The predicate interacts with the `YPA` by sending a request for registering the corresponding characteristics (`send_register_request/3`) and then it waits for the result of the interaction. Finally, it instantiates the term `Error` accordingly.

```
register(+Characteristics, +Ontology, -Error) :-
  send_register_request(+ypa, +Characteristics, +Ontology),
  get_answer(-Answer),
  ( Answer = done,
    Error = no_error
  ;
    Answer = failure,
    Error = error_registering_characteristics
  ).
```

Figure 5: A simplified version of the `register/3` predicate

In the case of the primitive `deregister/3`, it is similar to the register primitive, with the only difference that it will have exactly the opposite result. That is, it will eliminate characteristics or services from the `YPA`, that have been previously registered.

Finally, we proposed two primitives for searching for agents with specific characteristics:

- which_MASs(-List_Of_MASs, +Ontology, -Error)
- which_agents(-List_Of_Agents, +Characteristics, +Ontology, -Error)

The primitive `which_MASs` allows the agent to obtain the different MASs existing in the `YPA`, containing agents with similar characteristics. It will return the list of MASs in the first parameter, corresponding to the ontology specified in the second parameter. Since the implementation of this primitive and the one explained next is similar to the implementation of the predicate `register/3` (shown in Figure 5), it will be omitted.

In the case of the primitive `which_agents`, it allows the agents to acquire the information of all the registered agents having equal characteristics or providing equivalent services. Thus, this primitive will return the list of agents in the first parameter, having the characteristics specified in the second parameter and corresponding to the ontology defined in the third parameter.

Note that, although the primitive `which_agents` will only return the names of the agents, the CL will know the agents addresses, in order to be able to send messages. Thus, the CL also provides a set of primitives for the deliver and retrieval of messages, in order to allow the agent to interact with any other agents.

Whenever an agent tries to send a message to another agent, the CL will try to establish a connection with the corresponding agent directly and send the message. The CL may not be able to establish the connection with the agent, depending on different reasons. These possible reasons may be that the CL does not know the agent address or that the agent is not reachable at the location known by the CL. In either case, the CL will contact the `YPA` querying for the last address known of the agent and it will try again to send the message.

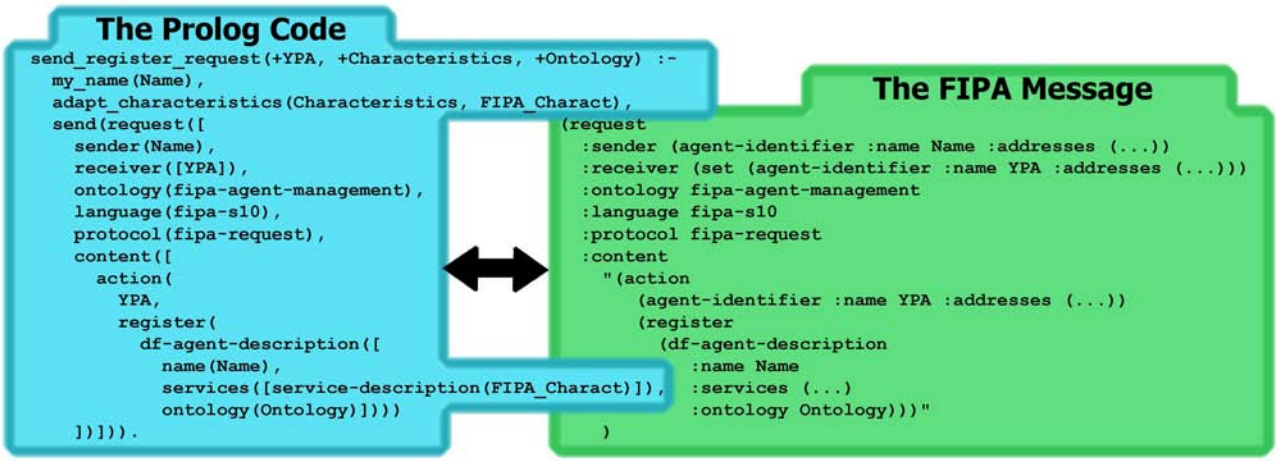


Figure 6: The `send_register_request` Predicate and the resulting FIPA Message

The CL uses the FIPA ACL Message Representation in String in order to allow agents developed using this framework to interact with any FIPA Complaint agent. Thus, it provides

a way of sending and receiving messages that allow the agent to exchange FIPA Messages. See for example the predicate `send_register_request` of Figure 5. The implementation of this predicate is shown in Figure 6, also showing the FIPA message that the CL will sent.

4 RELATED WORK

There are several agent platforms available for developing multi-agent systems [12]. Most of these platforms are focused either in the cognitive part of the architecture or in the infrastructural one. In other words, some platforms are FIPA-compliant concerning only the infrastructural problem, such as middle-ware issues, whereas other platforms are reasoning-centered, focusing on the behavior of a single agent.

JADE, a Java Agent Development Framework [2], is a software framework to develop agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. Similar to us, the goal of JADE is to simplify development while ensuring standard compliance through a comprehensive set of system services and agents. Thus, JADE can then be considered an agent middle-ware that implements an Agent Platform and a development framework, dealing with all those aspects that are not peculiar of the agent internals and that are independent of the applications.

However, there exist some differences between our approach and the one proposed by JADE. JADE offers a FIPA-compliant Agent Platform, including the AMS, the DF and the MTS, whereas our approach presents some variations to the FIPA Reference Model explained in detail in Section 2. While JADE has been fully coded in Java and an agent programmer should code his/her agents in Java, our framework is based on Logic Programming (LP), and the agents should be developed using Prolog.

Another approach corresponds to the MadKit Agent Platform Architecture [8], a generic multi-agent platform. This toolkit is based on a organizational model. It uses concepts of groups (similar to our MASs) and roles (similar to our agents characteristics or services) for agents to manage different agent models and multi-agent systems at the same time.

Unlike our approach, the MadKit architecture is based on a minimalist agent kernel decoupled from specific agency models. Thus the platform is not an agent platform in the classical sense. It presents the concept of "agent micro-kernel", handling the control of local groups and roles, the agent life-cycle management and the local message passing. In our approach, the `YPA` maintains the information of the different MASs and the agent characteristics and services, the agents are stand-alone programs and there is no agent life-cycle management, and finally, the CL provides message exchange between agents, independently of their location.

MadKit is focused only the infrastructural problem and it is not centered in the cognitive part of the architecture. In the case of JADE, it provides a full integration with JESS [7] offering a so-called JessBehaviour, whereas our framework, as we already mentioned, is implemented as an extension of LP, and thus, sophisticated Knowledge Representation and reasoning formalisms developed for LP can be easily used.

5 CONCLUSIONS AND FUTURE WORK

In this paper we have considered the development of agent interaction in Multi-Agent Systems. We have proposed a communication model among agents based on the FIPA Architecture and FIPA Interaction Protocols. Our main goal was to allow the agents both to reach each other by their characteristics and to exchange messages. In order to allow the interaction among heterogeneous agents, we have used standard Agent Communication Languages in the message exchange.

We have designed and implemented a framework as an extension of Prolog since this language is widely adopted for the development of intelligent agents. The framework corresponds to a set of primitives that follows the spirit of Logic Programming. Agents may register themselves in MASs in order to allow other agents to locate them easily. Any agent is allowed to search for agents having specific characteristics, or providing determined services. Thus, the framework proposed provide a reliable way for programming communicative agents without dealing with low-level details such as the actual location of an agent.

The resulting framework has some limitations that we have addressed as future work. One limitation concerns security aspects, for example, the `YPA` may allow the agents to create private MASs and restrict the access to specific agents. We are also planning to extend the framework adding implementations of the standards Interaction Protocols defined by FIPA. Another possible extension corresponds to agent mobility, since in our framework, agents identifies each other by their names, and this feature simplifies the implementation.

REFERENCES

[1] B. Bauer, J. P. Mller, and J. J. Odell. Agent UML: A Formalism for Specifying Multiagent Interaction. In *Agent Oriented Software Engineering,*. Ciancarini y Wooldridge (ed.), 2001.

[2] F. Bellifemine, A. Poggi, and G. Rimassa. JADE – A FIPA-compliant agent framework. In *Proceedings of the 4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'99)*, pages 97–108, London, UK, April 1999.

[3] T. Finin and Y. Labrou. Agent Communication Languages. In *Proceedings of ASA/MA'99, First International Symposium on Agent Systems and Applications, and Third International Symposium on Mobile Agents*, 1999.

[4] FIPA. Foundation for Intelligent Physical Agents. http://www.fipa.org.

[5] FIPA. Interaction Protocol Library Specification, November 2000.

[6] FIPA. Abstract Architecture Specification, December 2002.

[7] Ernest J. Friedman-Hill. *Jess, The Java Expert System Shell.* Sandia National Laboratories, Livermore, CA, USA, March 1998. Version 4.0.

[8] Olivier Gutknecht and Jacques Ferber. The MADKIT agent platform architecture. In Thomas Wagner and Omer F. Rana, editors, *Agents Workshop on Infrastructure for Multi-Agent Systems*, volume 1887 of *Lecture Notes in Computer Science*, pages 48–55. Springer, 2000.

[9] K. V. Hindriks, F. S. De Boer, Hoek Wiebe van der, and J. Jc Meyer. Agent programming in 3APL. *Autonomous Agents and Multi-Agent Systems*, 2(4):357–401, 1999. Publisher: Kluwer Academic Publishers, Netherlands.

[10] JACK. JACK Intelligent Agents Framework. http://www.agent-software.com/.

[11] Y. Labrou. Standardizing agent communication. In *Proceedings of the Advanced Course on Artificial Intelligence (ACAI'01)*. Springer-Verlag, 2001.

[12] Eleni Mangina. Review of Software Products for Multi-Agent Systems, 2002. http://www.agentlink.org/resources/software-report.html.

[13] Alexander Pokahr, Lars Braubach, and Winfried Lamersdorf. Jadex: A BDI reasoning engine. In Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah-Seghrouchni, editors, *Multi-Agent Programming*, volume 15 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, pages 149–174. Springer, 2005.

[14] M. Wooldridge. Semantic issues in the verification of agent communication languages. In *Journal of Autonomous Agents and Multi Agent Systems*, 2000.

A first approach to Abductive Defeasible Logic Programming: formalization and properties

Mauro J. Gómez Lucero Alejandro J. García Carlos I. Chesñevar Guillermo R. Simari
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Laboratorio de Investigación y Desarrollo de Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Bahía Blanca, B8000CPB, Argentina
{mjg, ajg, grs, cic}@cs.uns.edu.ar

Abstract

This article presents the confluence of two general ideas: Defeasible Logic Programming (DeLP, an argumentation based formalism for representing knowledge and reasoning) and Abduction in logic. In first place, we introduce a framework that formally states the problem of doing abduction (obtaining abductive explanations) in the particular case that the abductive theory is a DeLP Program. In the reminder of this work we face that problem, providing a formal characterization of the notion of abductive explanation, in such a way that we could then easily calculate the explanations from this formal characterization. An important advantage of this approach is that by proving the correctness of the characterization, we ensure the correctness of the method for obtaining explanations.

Keywords: Defeasible Reasoning, Abduction, Knowledge Representation and Reasoning.

1 INTRODUCTION

In this work we will present the confluence of two general ideas. Firstly, Defeasible Logic Programming (DeLP) is a formalism for representing knowledge and reasoning, that combines results of Logic Programming and Defeasible Argumentation. This formalism will be briefly introduced in section 2. Secondly, Abduction in logic represents a form of reasoning where, given a (logic) theory T and a sentence G, we try to find a set of sentences Φ (abductive explanation for G) such that $T \cup \Phi \vdash G$ and $T \cup \Phi$ is consistent.

Here, we face the problem of doing abduction in the particular case that the abductive theory is a (restricted) DeLP program $\mathcal{P}$, and the sentence to be explained is a literal in the language of $\mathcal{P}$. In order to formally state the problem, in section 3 we introduce the notion of Abductive-DeLP framework, and we present a notion of abductive explanation slightly adapted for that framework.

In particular, we will focus on the problem of obtaining abductive explanations (for a literal h in an Abductive-DeLP framework AF). The strategy adopted consists in providing a formal characterization of the notion of abductive explanation, in such a way that we could then easily calculate the explanations from this formal characterization. An important advantage of this approach is that by proving the correctness of the characterization (not done in this work for space reasons), we ensure the correctness of the method for obtaining explanations.

The formal characterization of the notion of explanation involves two parts. From the definition of abductive explanation, and considering the reasoning mechanism of DeLP, we know that the incorporation of an abductive explanation for a given literal to the abductive theory (DeLP program), causes

Partially financed by CONICET (PIP 5050), Universidad Nacional del Sur (PGI 24/ZN11) and Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096).

the *emergence* or *activation* of new arguments (and possibly, the deactivation of existent ones), which in turn causes the literal to be accepted. Then in section 4 we formalize which information should be present in an explanation (and which should not) in order to activate a given argument. On the other hand, in section 5 we formally state which arguments must the explanation activate and which must not in order to effectively explain the acceptance of the literal. Finally, combining both, we will have a formal characterization of the notion of abductive explanation.

2 DELP OVERVIEW

Defeasible Logic Programming (DeLP) combines results of Logic Programming and Defeasible Argumentation. The system is fully implemented and is available online [1]. A brief explanation is included below (see [2]for full details). A DeLP program $\mathcal{P}$ is a set of facts, strict rules and defeasible rules. *Facts* are ground literals representing atomic information or the negation of atomic information using strong negation "$\sim$" (*e.g.,* $chicken(little)$ or $\sim scared(little)$). *Strict Rules* represent non-defeasible information and are denoted $L_0 \leftarrow L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals (*e.g.,* $bird \leftarrow chicken$ or $\sim innocent \leftarrow guilty$). *Defeasible Rules* represent tentative information and are denoted $L_0 \prec L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals (*e.g.,* $\sim flies \prec chicken$ or $flies \prec chicken, scared$).

When required, $\mathcal{P}$ is denoted (Π, Δ) distinguishing the subset Π of facts and strict rules, and the subset Δ of defeasible rules (see Ex. 2.1). *Strong negation* is allowed in the head of rules, and hence may be used to represent contradictory knowledge. From a program (Π, Δ) contradictory literals could be derived. Nevertheless, the set Π (which stands for non-defeasible information) must possess certain internal coherence, *i.e.*, no pair of contradictory literals can be derived from Π.

A defeasible rule is used to represent tentative information that may be used if nothing could be posed against it. Observe that strict and defeasible rules are ground. However, following the usual convention [3], some examples use "schematic rules" with variables.

Example 2.1. *Consider the DeLP program* $(\Pi_{2.1}, \Delta_{2.1})$ *where:*

$$\Pi_{2.1} = \left\{ \begin{array}{l} bird(X) \leftarrow chicken(X) \\ chicken(tina) \\ chicken(little) \\ scared(tina) \end{array} \right\} \quad \Delta_{2.1} = \left\{ \begin{array}{l} flies(X) \prec bird(X) \\ flies(X) \prec chicken(X), scared(X) \\ \sim flies(X) \prec chicken(X) \end{array} \right\}$$

This program has three defeasible rules representing tentative information about the flying ability of birds in general, and about regular chickens and scared ones. It also has a strict rule expressing that every chicken is a bird, and three facts: 'tina' and 'little' are chickens, and 'tina' is scared.

Given a DeLP program $\mathcal{P}$, a *defeasible derivation* for a literal L from $\mathcal{P}$ is a (tentative) proof for L involving strict and defeasible rules in the program. The reason for using the term defeasible is that it is possible to derive contradictory literals from $\mathcal{P}$. For example, from $(\Pi_{2.1}, \Delta_{2.1})$ in Ex. 2.1 it is possible to derive $flies(tina)$ and $\sim flies(tina)$.

For the treatment of contradictory knowledge DeLP incorporates a defeasible argumentation formalism. This formalism allows the identification of those pieces of knowledge that are in contradiction, and a *dialectical process* is used for deciding which information prevails as warranted. This dialectical process (see below) involves the construction and evaluation of arguments that either support or interfere with the query under analysis. As we will show next, arguments supporting the answer for a given query will be shown in a particular way using *dialectical trees*. The definition of dialectical tree will be included below, but first, we will give a brief explanation of other related concepts (for the details see [2]).

Definition 2.1 (Argument Structure). *Let (Π, Δ) be a DeLP program. . We will say that $\langle \mathcal{A}, \alpha \rangle$ is an argument structure (or just argument) for a literal α from (Π, Δ), if $\mathcal{A}$ is the minimal set of defeasible rules ($\mathcal{A} \subseteq \Delta$), such that: (1) there exists a defeasible derivation for α from $\Pi \cup \mathcal{A}$, and (2) the set $\Pi \cup \mathcal{A}$ is non-contradictory.*

Example 2.2. *From the DeLP program $(\Pi_{2.1}, \Delta_{2.1})$ the following arguments can be obtained:*
$\langle \mathcal{A}_1, flies(tina) \rangle$, *where* $\mathcal{A}_1 = \{flies(tina) \prec bird(tina)\}$
$\langle \mathcal{A}_2, \sim flies(tina) \rangle$, *where* $\mathcal{A}_2 = \{\sim flies(tina) \prec chicken(tina)\}$
$\langle \mathcal{A}_3, flies(tina) \rangle$, *where* $\mathcal{A}_3 = \{flies(tina) \prec chicken(tina), scared(tina)\}$

A literal L is *warranted* if there exists a non-defeated argument $\mathcal{A}$ supporting L. To establish if $\langle \mathcal{A}, \alpha \rangle$ is a non-defeated argument, *defeaters* for $\langle \mathcal{A}, \alpha \rangle$ are considered, *i.e.*, counter-arguments that by some criterion are preferred to $\langle \mathcal{A}, \alpha \rangle$. In DeLP, the comparison criterion is usually *generalized specificity*, but in the examples given in this paper we will abstract away from this particular criterion, since in this work it introduces unnecessary complications. Since defeaters are arguments, there may exist defeaters for them, and defeaters for these defeaters, and so on. Thus, a sequence of arguments called *dialectical line* is constructed, where each argument defeats its predecessor. In order to avoid undesirable sequences that may represent circular or fallacious argumentation lines, in DeLP a dialectical line is *acceptable* if it satisfies certain constraints (see [2]).

Example 2.3. *From Ex. 2.2, we have that argument $\langle \mathcal{A}_2, \sim flies(tina) \rangle$ defeats $\langle \mathcal{A}_1, flies(tina) \rangle$, argument $\langle \mathcal{A}_3, flies(tina) \rangle$ is a defeater for $\langle \mathcal{A}_2, \sim flies(tina) \rangle$, and the sequence of arguments $[\langle \mathcal{A}_1, flies(tina) \rangle, \langle \mathcal{A}_2, \sim flies(tina) \rangle, \langle \mathcal{A}_3, flies(tina) \rangle]$ is an acceptable argumentation line.*

Clearly, there might be more than one defeater for a particular argument. Therefore, many acceptable argumentation lines could arise from a given argument, leading to a tree structure. Before defining the notion of *dialectical tree*, we need to introduce some terminology. Let $\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle]$ be an acceptable argumentation line, and $\langle \mathcal{B}_i, q_i \rangle$ be a defeater of $\langle \mathcal{A}_n, h_n \rangle$ such that $\Lambda' = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle, \langle \mathcal{B}_i, q_i \rangle]$ is also acceptable. Then we will say that $\langle \mathcal{B}_i, q_i \rangle$ *extends* Λ.

Definition 2.2 (Dialectical tree [2]). *Let $\langle \mathcal{A}_0, h_0 \rangle$ be an argument from a program $\mathcal{P}$. A dialectical tree for $\langle \mathcal{A}_0, h_0 \rangle$ from $\mathcal{P}$, denoted $\mathcal{T}_{\mathcal{P}}(\langle \mathcal{A}_0, h_0 \rangle)$, is defined as follows:*

(1) The root of the tree is labelled with $\langle \mathcal{A}_0, h_0 \rangle$.
(2) Let N be a node of the tree labelled $\langle \mathcal{A}_n, h_n \rangle$, and $\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle]$ be the (acceptable) argumentation line corresponding to the sequence of labels of the path from the root to N. Let $\{ \langle \mathcal{B}_1, q_1 \rangle, \langle \mathcal{B}_2, q_2 \rangle, \ldots, \langle \mathcal{B}_k, q_k \rangle \}$ be the set of all arguments extending Λ. Then for each argument $\langle \mathcal{B}_i, q_i \rangle$, the node N has a child N_i labelled $\langle \mathcal{B}_i, q_i \rangle$. If there exist no argument extending Λ, then N is a leaf.

In a dialectical tree, every node (except the root) represents a defeater of its parent, and leaves correspond to non-defeated arguments. Each path from the root to a leaf corresponds to a different acceptable argumentation line. A dialectical tree provides a structure for considering all the possible acceptable argumentation lines that can be generated for deciding whether an argument is ultimately defeated. We call this tree *dialectical* because it represents an exhaustive dialectical analysis for the argument in its root.

Given a literal h and an argument $\langle \mathcal{A}, h \rangle$ from a program $\mathcal{P}$, to decide whether a literal h is warranted, every node in the dialectical tree $\mathcal{T}_{\mathcal{P}}(\langle \mathcal{A}, h \rangle)$ is recursively marked as "D" (*defeated*) or "U" (*undefeated*), obtaining a marked dialectical tree $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h \rangle)$ as follows:

1. All leaves in $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h\rangle)$ are marked as "U"s, and
2. Let N be an inner node of $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h\rangle)$. Then N will be marked as "U" iff every child of N is marked as "D". The node N will be marked as "D" iff it has at least a child marked as "U".

Given an argument $\langle \mathcal{A}, h\rangle$ obtained from $\mathcal{P}$, if the root of $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h\rangle)$ is marked as "U", then we will say that $\mathcal{T}^*_{\mathcal{P}}(\langle \mathcal{A}, h\rangle)$ *warrants* h and that h is *warranted* from $\mathcal{P}$ (that will be noted $\mathcal{P} \mid\sim_{W} h$).

In this paper, an argument $\langle \mathcal{A}, h\rangle$ will be depicted as a triangle, whose upper vertex is labelled with the conclusion h, and the set of defeasible rules $\mathcal{A}$ are associated with the triangle itself. Marked dialectical trees will be depicted as shown in figure 1.

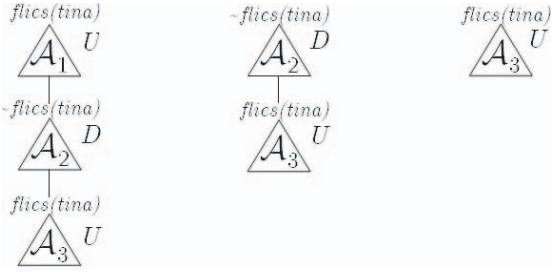


Figure 1: Dialectical trees for $flies(tina)$.

Example 2.4. *(Extends Ex. 2.3) Figure 1 shows the marked dialectical tree* $\mathcal{T}^*_{\mathcal{P}\ 2.1}(\langle \mathcal{A}_1, f\rangle)$ *(the leftmost tree), which has only one argumentation line. Observe that the argument* $\langle \mathcal{A}_2, \sim f\rangle$ *interferes with the warrant of 'flies(tina)' and the argument* $\langle \mathcal{A}_3, f\rangle$ *reinstates* $\langle \mathcal{A}_1, f\rangle$*. The root of* $\mathcal{T}^*_{\mathcal{P}\ 2.1}(\langle \mathcal{A}_1, f\rangle)$ *is marked as "*U*" and therefore the literal 'flies(tina)' is warranted.*

3 ABDUCTIVE DELP FRAMEWORK

In this section we will introduce a framework for doing abduction in a (restricted) version of DeLP without strict rules. Before, we need to introduce certain distinguished sets of literals associated with any set of defeasible rules. Let Δ be a set of defeasible rules. Then we will use $all\_lits(\Delta)$ to denote the set of all literals appearing in defeasible rules in Δ. Also, we will use $head\_lits(\Delta)$ to denote the set of all literals that are heads of defeasible rules in Δ (*i.e.*, $head\_lits(\Delta) = \{H | H -\!\prec B_1, B_2, ..., B_n \in \Delta\}$). Finally, we will use $body\_lits(\Delta)$ to denote the set of all literals that appear in the body of a defeasible rule in Δ. (*i.e.*, $body\_lits(\Delta) = \{B_i | H -\!\prec B_1, ..., B_i, ..., B_n \in \Delta\}$).

Definition 3.1 (A-DeLP Framework). *An A-DeLP framework is a pair* $\langle \Pi_f \cup \Delta, A\rangle$ *where*

- $\Pi_f \cup \Delta$*, the theory, is a DeLP program without strict rules, ie.* Π_f *is a consistent set of literals (called facts) and* Δ *is a set of defeasible rules.*
- *A is a set of distinguished literals in the language of* $\Pi_f \cup \Delta$*, called abducibles, such that*

$$body\_lits(\Delta) \backslash (heads(\Delta) \cup \Pi_f \cup \overline{\Pi_f}) \subseteq A \subseteq (all\_lits(\Delta) \cup \overline{all\_lits(\Delta)}) \backslash (\Pi_f \cup \overline{\Pi_f})$$

Given a literal Q, an abductive explanation for Q w.r.t. the abductive framework, is defined as a set of abducibles Φ ($\Phi \subseteq A$) such that:

- $\Pi_f \cup \Delta \cup \Phi \mid\sim_{W} Q$ and
- $\Pi_f \cup \Phi$ is consistent.

The set A associated with an A-DeLP framework constitutes the domain of the explanations in that framework, *i.e.,* A is the set of literals (abducibles) that can integrate the explanations. The set A must be specified when defining the A-DeLP framework and it could be any set of literals satisfying the restrictions stated by the previous definition. The election of a specific set of abducibles A depends on the application domain.

The adjective *abductive* is used to distinguish explanations in the abductive sense from the notion of explanation in the context of DeLP (or dialectical explanations), where dialectical trees constructed to answer a query can be seen as explanations for the answer obtained. From now on, when we use the term explanation we mean abductive explanation.

After introducing the elements for performing abduction in DeLP, we have to deal with the problem of obtaining explanations. In what follows, we will present a formal characterization of the notion of abductive explanation, in such a way that we could then easily calculate the explanations from this formal characterization.

4 CHARACTERIZING THE NOTION OF (POTENTIAL) ARGUMENT ACTIVATION

As we have seen in section 2, the (state of) warrant of a given literal w.r.t. a DeLP program depends on the arguments that can be computed from that program. Then in order to consider a set Φ of abducibles as an explanation of the warrant of a given literal, Φ must cause the *activation* of certain arguments and, as we will see later, the *non activation* of others. In this section, we formally state the conditions that a given set Φ of abducibles must satisfy in order to cause the activation of a certain argument. More precisely, those conditions specify the presence of certain literals in Φ and the absence of others.

As we have seen in section 2, the notion of argument is defined with respect to a *concrete* DeLP program. On the other hand, when performing abduction we work with an *incomplete* DeLP program, trying to find a set of facts (more precisely abducibles) to *complete* the DeLP program in order to warrant a given literal. Then the notion of argument we introduced in section 2 is not suitable in the context of abduction, and consequently we need to introduce a notion of *potential* argument. For the next definition, we will use $\mathcal{L}(AF)$ to denote the language associated with AF.

Definition 4.1 (Potential Argument). *Let $AF = \langle \Pi_f \cup \Delta, A\rangle$ be an A-DeLP framework. Let $Q \in \mathcal{L}(AF)$. Let $\mathcal{B} \subseteq \Delta$. Then $\langle\langle \mathcal{B}, Q\rangle\rangle$ is a potential argument w.r.t. AF iff there exists a non contradictory set of literals $\Omega \subseteq \mathcal{L}(AF)$ such that $\langle \mathcal{B}, Q\rangle$ is an argument w.r.t. (Ω, Δ).*

Example 4.1. *Consider an A-DeLP framework $AF_1 = \langle \Pi^1_f \cup \Delta_1, A_1\rangle$, where:*
$\Pi^1_f = \{b, \sim g\}$,
$\Delta_1 = \{(a \prec b, c), (c \prec d, e), (f \prec g), (g \prec h)\}$
$A_1 = \{c, \sim c, d, e, h\}$

Then the following are potential arguments w.r.t. AF_1:

$\langle\langle\{(a \prec b, c), (c \prec d, e)\}, a\rangle\rangle$ *(assuming for example $\Omega = \{b, d, e\}$)*

$\langle\langle\{(a \prec b, c)\}, a\rangle\rangle$ *(assuming for example $\Omega = \{b, c\}$)*

$\langle\langle\emptyset, g\rangle\rangle$ *(assuming for example $\Omega = \{g\}$)*

On the other hand, $\langle\langle\{(a \prec b, c), (f \prec g)\}, a\rangle\rangle$ is not a potential argument w.r.t. AF, since $\langle\{(a \prec b, c), (f \prec g)\}, a\rangle$ will not be an argument w.r.t. (Ω, Δ_1), for any Ω considered. Note that the defeasible rule $f \prec g$ will never be necessary to defeasibly derive a, violating the minimality condition in the definition of argument.

The following definition formally introduces the notion of (potential) argument activation, that was informally used at the beginning of this section.

Definition 4.2 (Potential Argument Activation). *Let $AF = \langle \Pi_f \cup \Delta, A \rangle$ be an A-DeLP framework. Let $\langle\langle \mathcal{B}, Q \rangle\rangle$ be a potential argument w.r.t. AF and let $\Phi \subseteq A$. We say that $\langle\langle \mathcal{B}, Q \rangle\rangle$ is activated by Φ in AF (or alternatively that Φ activates $\langle\langle \mathcal{B}, Q \rangle\rangle$) iff $\Phi \nvdash \bot$ and $\langle \mathcal{B}, Q \rangle$ is an argument w.r.t. $(\Pi_f \cup \Phi, \Delta)$.*

Example 4.2. *Consider the A-DeLP framework AF_1 of example 4.1. Let $\langle\langle \mathcal{B}, a \rangle\rangle$ be a potential argument w.r.t. AF_1, where $\mathcal{B}$=$\{(a \prec b, c), (c \prec d, e)\}$. The set of abducibles $\Phi_1 = \{d, e, h\}$ activates $\langle\langle \mathcal{B}, a \rangle\rangle$, because $\langle \mathcal{B}, a \rangle$ is an argument w.r.t. $(\Pi^1_f \cup \Phi_1, \Delta_1)$. As can be seen in figure 2a, Φ_1 completes Π^1_f with those additional facts needed to defeasibly derive a (using rules in $\mathcal{B}$). Furthermore, it can be noted that the abducible $h \in \Phi_1$ does not contribute to the activation of $\langle\langle \mathcal{B}, a \rangle\rangle$. Note that $\Phi_2 = \Phi_1 \setminus \{h\} = \{d, e\}$ also activates $\langle\langle \mathcal{B}, a \rangle\rangle$.*

On the other hand, the set $\Phi_3 = \{d\}$ does not activate $\langle\langle \mathcal{B}, a \rangle\rangle$ (see figure 2b). Note that e does not belong to $\Pi^1_f \cup \Phi_3$, and it is necessary to defeasibly derive a.

The set $\Phi_4 = \{d, e, \sim c\}$ does also not activate $\langle\langle \mathcal{B}, a \rangle\rangle$ (see figure 2c), this time because Φ_4 incorporates a fact ($\sim$c) that contradicts an intermediate conclusion *of $\langle\langle \mathcal{B}, a \rangle\rangle$, violating the consistency condition in the definition of argument.*

Another interesting case is $\Phi_5 = \{d, e, c\}$, which does not activate $\langle\langle \mathcal{B}, a \rangle\rangle$ (see figure 2d). Note that Φ_5 incorporates c, which coincides with the head of $c \prec d, e$, one of the rules in $\mathcal{B}$. In such case, $c \prec d, e$ is unnecessary for deriving a, because as c is a fact, we can directly introduce it in the derivation. So $\mathcal{B}$ is not minimal for deriving a, and then violates the minimality condition of the definition of argument. Now, consider the potential argument $\langle\langle \mathcal{C}, a \rangle\rangle$, where $\mathcal{C} = \{a \prec b, c\}$ (i.e., $\mathcal{C}$ is $\mathcal{B}$ without $c \prec d, e$). Then it holds that Φ_5 effectively activates $\langle\langle \mathcal{C}, a \rangle\rangle$ (see figure 2e).

The set $\Phi_6 = \{e\}$ activates the empty potential argument $\langle\langle \emptyset, e \rangle\rangle$. In general, any set of abducibles Φ containing a literal Q activates the empty potential argument $\langle\langle \emptyset, Q \rangle\rangle$.

Finally, consider the potential argument $\langle\langle \mathcal{D}, f \rangle\rangle$, where $\mathcal{D} = \{(f \prec g), (g \prec h)\}$, and let $\Phi_7 = \{h\}$. Then Φ_7 does not activate $\langle\langle \mathcal{D}, f \rangle\rangle$ because of an inconsistency involving an intermediate conclusion and a fact (as previously showed with Φ_4 and $\langle\langle \mathcal{B}, a \rangle\rangle$), but this time, the fact causing the inconsistency, $\sim g$, belongs to Π^1_f (see figure 2f). So, there exists no set of abducibles activating $\langle\langle \mathcal{D}, f \rangle\rangle$.

Next, we will introduce a property which states certain necessary and sufficient conditions for a potential argument $\langle\langle \mathcal{B}, Q \rangle\rangle$ to be activated by a set of abducibles Φ. More precisely, the property states which literals must Φ contain (and which must not) in order to activate $\langle\langle \mathcal{B}, Q \rangle\rangle$. Before introducing the property, we need to define two distinguished sets of literals associated with any potential argument.

Definition 4.3. *Let $\langle\langle \mathcal{B}, Q \rangle\rangle$ be a potential argument. We define two sets of literals associated with $\langle\langle \mathcal{B}, Q \rangle\rangle$:*

- $heads(\langle\langle \mathcal{B}, Q \rangle\rangle) =_{def} head\_lits(\mathcal{B})$
- $base(\langle\langle \mathcal{B}, Q \rangle\rangle) =_{def} \begin{cases} \{Q\}, & \text{if } \mathcal{B} = \emptyset; \\ all\_lits(\mathcal{B}) \setminus head\_lits(\mathcal{B}), & \text{otherwise.} \end{cases}$

The reason for using of the term *base* becomes evident when considering the graphical representation we adopted for potential arguments in figure 4.2. That is, as a triangle with the conclusion on the top, and expliciting inside the defeasible rules conforming it chained in a top-down derivation. The following example, and the figures associated with it, illustrates what we are saying.

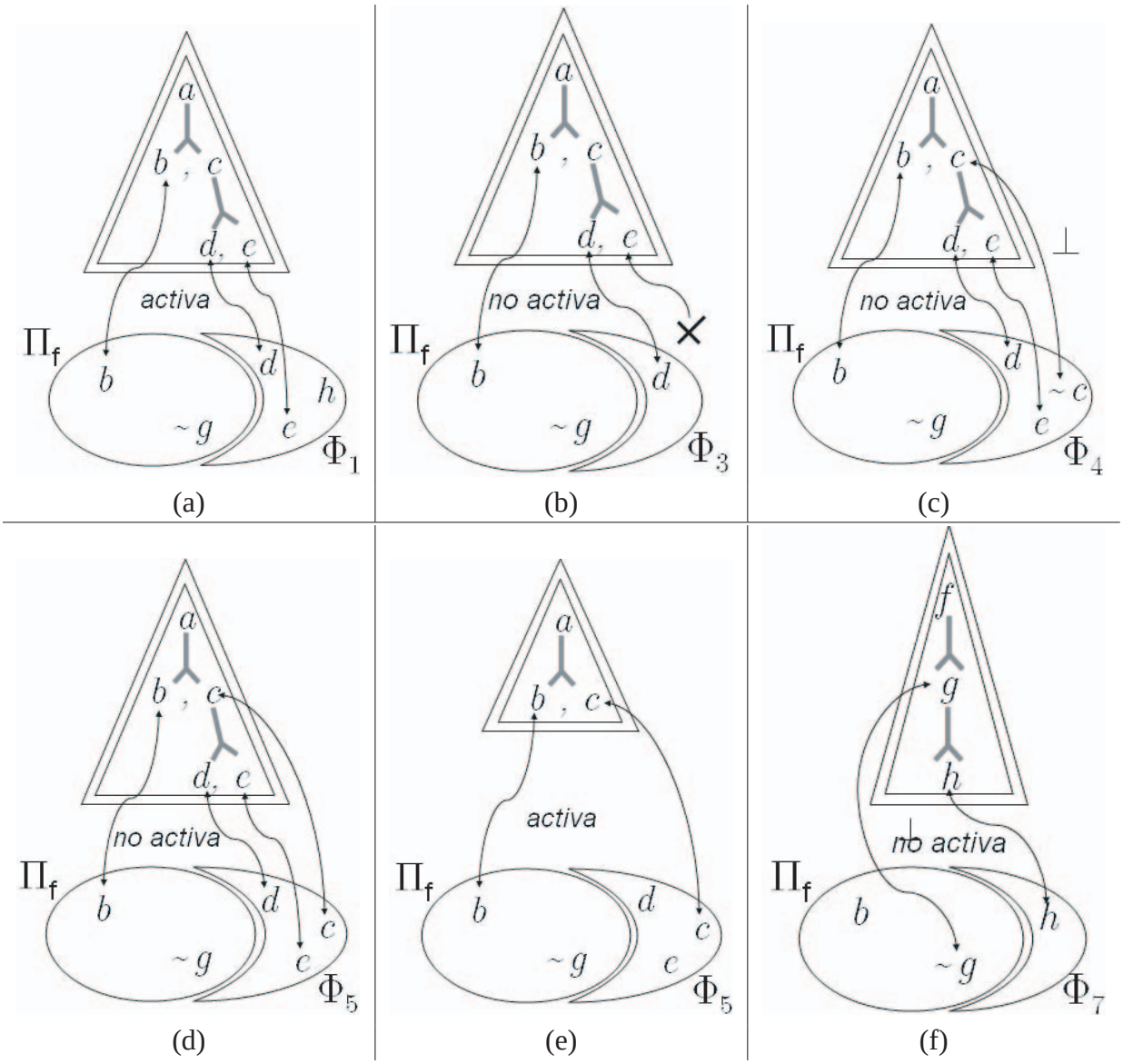


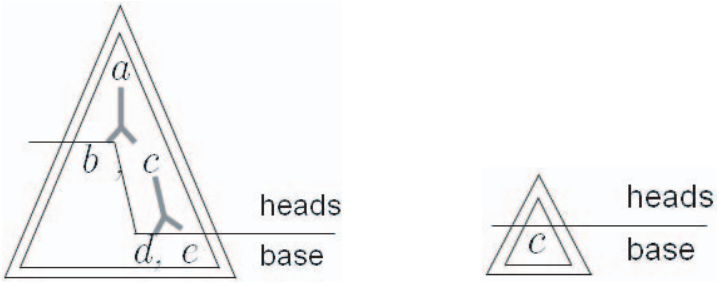
Figure 2: Potential Argument Activation

Example 4.3. *Consider the A-DeLP framework AF_1 of example 4.1. Let $\langle\langle \mathcal{B}, a\rangle\rangle$ be the potential argument of example 4.2 ($\mathcal{B}$=$\{(a \prec b, c)$, $(c \prec d, e)\}$). Then $heads(\langle\langle \mathcal{B}, a\rangle\rangle) = \{a, c\}$ and $base(\langle\langle \mathcal{B}, a\rangle\rangle) = \{b, d, e\}$ (see figure 3a). Consider the empty potential argument $\langle\langle \emptyset, c\rangle\rangle$. Then $heads(\langle\langle \emptyset, c\rangle\rangle) = \emptyset$ and $base(\langle\langle \emptyset, c\rangle\rangle) = \{c\}$ (see figure 3b).*

Property 4.1. *Let $AF = \langle \Pi_f \cup \Delta, A\rangle$ be an A-DeLP framework. Let $\langle\langle \mathcal{B}, Q\rangle\rangle$ be a potential argument w.r.t. AF and let $\Phi \subseteq A$. Then Φ activates $\langle\langle \mathcal{B}, Q\rangle\rangle$ iff $\Phi \nvdash \bot$ and the following conditions hold:*

- $base(\langle\langle \mathcal{B}, Q\rangle\rangle) \subseteq \Pi_f \cup \Phi$
- *heads($\langle\langle \mathcal{B}, Q\rangle\rangle$)* $\cap (\Pi_f \cup \Phi) = \emptyset$
- $\overline{heads(\langle\langle \mathcal{B}, Q\rangle\rangle)} \cap (\Pi_f \cup \Phi) = \emptyset$

It can easily be verified that the property holds for each case considered in example 4.2. In particular, consider the potential argument $\langle\langle \mathcal{D}, f\rangle\rangle$ defined in example 4.2 ($\mathcal{D}$= $\{(f \prec e)$, $(e \prec g)\}$).



$\langle\langle \mathcal{B}, a \rangle\rangle$ (a) $\langle\langle \emptyset, c \rangle\rangle$ (b)

Figure 3: base and heads sets of a potential argument.

As we concluded before, there exists no set of abducibles activating $\langle\langle \mathcal{D}, f \rangle\rangle$. In fact, note that $g \in heads\langle\langle \mathcal{D}, f \rangle\rangle$ and $\sim g \in \Pi_f^1$, and then the second condition in the previous property is violated independently of the set Φ of abducibles considered.

As we have already shown in the previous examples, given an A-DeLP framework, there may exist potential arguments that could not be activated in such framework. That is the same to say, as we have just seen, that they violate at least one of the three conditions of property 4.1, for any set Φ of abducibles considered. So, given an A-DeLP framework, we could ignore such *non activable* potential argument, concentrating only on the others, that we will call *activable* potential arguments. Next, we will formally define the notion of activable potential argument.

Definition 4.4 (Activable Potential Argument). *Let $AF = \langle \Pi_f \cup \Delta, A \rangle$ be an A-DeLP framework. Let $\langle\langle \mathcal{B}, Q \rangle\rangle$ be a potential argument w.r.t. AF. We say that $\langle\langle \mathcal{B}, Q \rangle\rangle$ is activable in AF iff there exists a subset Φ of A such that Φ activates $\langle\langle \mathcal{B}, Q \rangle\rangle$.*

Now that we have formally defined the notion of activable potential argument, we will present, through the next property, a more practical (directly implementable) characterization of that notion. More precisely, it states the restrictions that must satisfy a given potential argument, in terms of it´s base and heads sets, in order to be activable.

Property 4.2. *Let $AF = \langle \Pi_f \cup \Delta, A \rangle$ be an A-DeLP framework. Let $\langle\langle \mathcal{B}, Q \rangle\rangle$ be a potential argument w.r.t. AF. Then $\langle\langle \mathcal{B}, Q \rangle\rangle$ is activable in AF (noted as $\langle\langle \mathcal{B}, Q \rangle\rangle^+$) iff it satisfies the following conditions:*

- $base(\langle\langle \mathcal{B}, Q \rangle\rangle) \subseteq \Pi_f \cup A$
- $heads(\langle\langle \mathcal{B}, Q \rangle\rangle) \cap \Pi_f = \emptyset$
- $\overline{heads(\langle\langle \mathcal{B}, Q \rangle\rangle)} \cap \Pi_f = \emptyset$

Consider the three arguments that appear in example 4.2. Note that the potential argument $\langle\langle \mathcal{D}, f \rangle\rangle$ ($\mathcal{D} = \{(f \prec e), (e \prec g)\}$), which was identified as non-activable, violates indeed the third condition of this property. The other two, $\langle\langle \mathcal{B}, a \rangle\rangle$ ($\mathcal{B}$=$\{(a \prec b, c)$, $(c \prec d, e)\}$) and $\langle\langle \mathcal{C}, a \rangle\rangle$, ($\mathcal{C} = \{a \prec b, c\}$), are activable (see figures 2a and 2e respectively) and then as it can be easily verified, they both satisfy the three conditions of the property.

Finally, we will present a new version of property 4.1, specific for activable potential arguments.

Property 4.3. *Let $AF = \langle \Pi_f \cup \Delta, A \rangle$ be an A-DeLP framework. Let $\langle\langle \mathcal{B}, Q \rangle\rangle^+$ be an activable potential argument w.r.t. AF and let $\Phi \subseteq A$. Then Φ activates $\langle\langle \mathcal{B}, Q \rangle\rangle^+$ iff $\Phi \nvdash \bot$ and the following conditions are satisfied:*

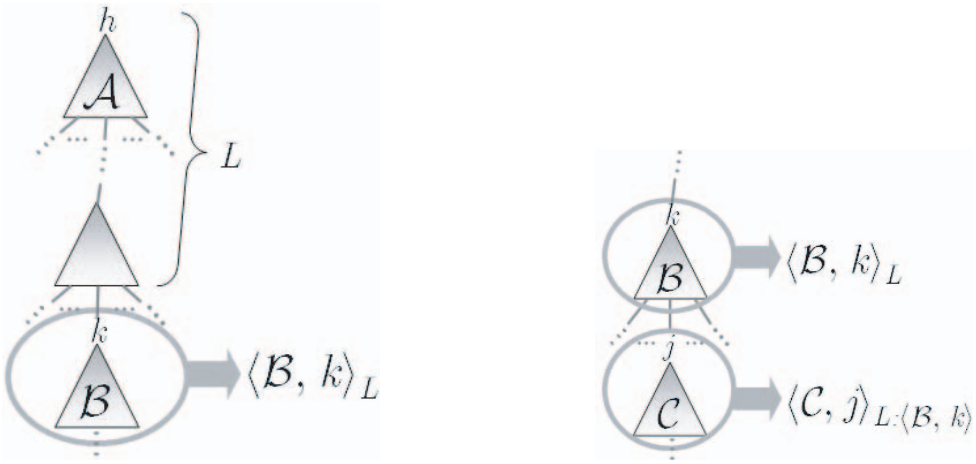
- $base(\langle\langle \mathcal{B}, Q \rangle\rangle^+) \setminus \Pi_f \subseteq \Phi$
- $heads(\langle\langle \mathcal{B}, Q \rangle\rangle^+) \cap \Phi = \emptyset$
- $\overline{heads(\langle\langle \mathcal{B}, Q \rangle\rangle^+)} \cap \Phi = \emptyset$

5 CHARACTERIZING WARRANT EXPLANATIONS

As we have said in section 3, our purpose is to provide a formal characterization of the notion of abductive explanation, in such a way that we could then easily calculate the explanations from this formal characterization. In the previous section, we presented partially this characterization, focusing only on the conditions under which a given set Φ of abducibles activates a given (activable) potential argument. In this section we will complete the characterization. More specifically, we will provide a logical formalization in First Order Logic that captures the notion of which (activable) potential arguments must a given set Φ of abducibles activate, and which must not, in order to explain the warrant of a given literal h.

As a first step, we will define a specific notation for denoting nodes in a dialectical tree. At first sight, it seems that to univocally identify a certain node it is sufficient to name the argument labelling it. But that is wrong because two nodes in a dialectical tree could be labelled with the same argument. An effective way of identifying a node is making reference not only to the argument labelling it, but also to the argumentation line conformed by the (arguments labelling the) ancestors of the node in the tree.

Consider a dialectical tree $\mathcal{T}(\langle \mathcal{A}, h \rangle)$, let N be a node of $\mathcal{T}(\langle \mathcal{A}, h \rangle)$ labeled with $\langle \mathcal{B}, k \rangle$ and let L be the argumentation line corresponding to the path from the root of $\mathcal{T}(\langle \mathcal{A}, h \rangle)$ to the father of N inclusively. Then we will use $\langle \mathcal{B}, k \rangle_L$, read as $\langle \mathcal{B}, k \rangle$ with ancestors line L, to refer to the node N (see figure 4a). Additionally, let $L = [\langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, ..., \langle \mathcal{A}_n, h_n \rangle]$ be an arbitrary argumenta-



(a) (b)

Figure 4: Notation for nodes in a dialectical tree.

tion line, and let $\langle \mathcal{B}, k \rangle$ be an argument. Then will use $L : \langle \mathcal{B}, k \rangle$ to denote the argumentation line $[\langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, ..., \langle \mathcal{A}_n, h_n \rangle, \langle \mathcal{B}, k \rangle]$, that results from adding $\langle \mathcal{B}, k \rangle$ at the end of L. Finally, if $\langle \mathcal{B}, k \rangle_L$ denotes a node and it has a child labeled with $\langle \mathcal{C}, j \rangle$, then combining both notations introduced, we can use $\langle \mathcal{C}, j \rangle_{L:\langle \mathcal{B},k \rangle}$ to denote the child node (see figure 4b).

Next we will present the analysis that lead us to the first logic formula. Let $AF = \langle \mathcal{P}, A \rangle$ be an A-DeLP framework, let $\Phi \subseteq A$ be a consistent set of abducibles and let h be a literal. Then from the definition of warrant presented in Section 2 and from our previous discussion we can state the following:

h will be warranted w.r.t. the DeLP program $\mathcal{P} \cup \Phi$ (1)

iff

there exist an argument $\langle \mathcal{A}, h \rangle$ in $\mathcal{P} \cup \Phi$ (2)

and

$\mathcal{T}_{(\mathcal{P} \cup \Phi)}(\langle \mathcal{A}, h \rangle)$, the dialectical tree for $\langle \mathcal{A}, h \rangle$, has its root undefeated.

But the statement (1) is the same to say that Φ *will be an explanation for the warrant of* h. Also, the statement (2) is in turn the same to say that *there exist an activable potential argument* $\langle\langle \mathcal{A}, h \rangle\rangle^+$ *in* AF *such that* Φ *activates* $\langle\langle \mathcal{A}, h \rangle\rangle^+$. By rephrasing statements (1) and (2) into the equivalent formulation given before, we can define the notion of explanation as follows:

Φ will be an explanation for the warrant of h ***iff*** *there exists an activable potential argument $\langle\langle \mathcal{A}, h \rangle\rangle^+$ in AF* ***such that*** *Φ activates $\langle\langle \mathcal{A}, h \rangle\rangle^+$* ***and*** *$\mathcal{T}_{(\mathcal{P} \cup \Phi)}(\langle \mathcal{A}, h \rangle)$ has its root undefeated.*

Let $PotArgs^+_{AF}$ be the set of all activable potential arguments w.r.t. AF. Then the previous sentence is captured by the following formula:

$$exp\_warrant(\Phi, h) =_{def} \bigvee_{\langle\langle \mathcal{A},h \rangle\rangle^+ \in PotArgs^+_{AF}} act(\Phi, \langle\langle \mathcal{A}, h \rangle\rangle^+) \wedge U(\Phi, \langle \mathcal{A}, h \rangle_{[]}) \qquad (\alpha)$$

Now, consider an argument $\langle \mathcal{A}, h \rangle$ w.r.t. $\mathcal{P} \cup \Phi$, let $\mathcal{T}_{(\mathcal{P} \cup \Phi)}(\langle \mathcal{A}, h \rangle)$ be the dialectical tree for $\langle \mathcal{A}, h \rangle$ and let $\langle \mathcal{N}, k \rangle_L$ be a node of $\mathcal{T}_{(\mathcal{P} \cup \Phi)}(\langle \mathcal{A}, h \rangle)$. Then according to the criterion used for marking a dialectical tree (section 2), we can say that:

$\langle \mathcal{N}, k \rangle_L$ will be defeated ***iff*** *there exists a child $\langle \mathcal{M}, q \rangle_{L:\langle \mathcal{N},k \rangle}$ of $\langle \mathcal{N}, k \rangle_L$* (3)
such that *$\langle \mathcal{M}, q \rangle_{L:\langle \mathcal{N},k \rangle}$ is undefeated.*

But, by definition of dialectical tree, the node $\langle \mathcal{N}, k \rangle_L$ has a child $\langle \mathcal{M}, q \rangle_{L:\langle \mathcal{N},k \rangle}$ iff there exists an argument $\langle \mathcal{M}, q \rangle$ w.r.t. $\mathcal{P} \cup \Phi$ which extends the argumentation line $L : \langle \mathcal{N}, k \rangle$. Moreover, that is equivalent to say that Φ activates some potential argument $\langle\langle \mathcal{M}, q \rangle\rangle^+$ w.r.t. AF which extends $L : \langle \mathcal{N}, k \rangle$. Below is the statement that results from transcribing (3) according to de last analysis:

$\langle \mathcal{N}, k \rangle_L$ will be defeated ***iff*** *there exists an activable potential argument $\langle\langle \mathcal{M}, q \rangle\rangle^+$ w.r.t. AF which extends $L : \langle \mathcal{N}, k \rangle$* ***such that*** *$\Phi$ activates $\langle\langle \mathcal{M}, q \rangle\rangle^+$* ***and*** *$\langle \mathcal{M}, q \rangle_{L:\langle \mathcal{N},k \rangle}$ is undefeated.*

OBSERVATION: We could restrict $\langle\langle \mathcal{M}, q \rangle\rangle^+$ to be a non-empty argument without changing the meaning of the sentence. The reason is that if $\mathcal{M} = \emptyset$, we can ensure that $\langle\langle \mathcal{M}, q \rangle\rangle^+$ will not be activated by Φ [1].

Let L be an argumentation line. We will use $PotArgsExt^+_{AF}(L)$ to denote the set of all non-empty activable potential arguments w.r.t. AF which extend L. Then the previous sentence is captured by the following formula:

$$D(\Phi, \langle \mathcal{N}, k \rangle_L) =_{def} \bigvee_{\langle\langle \mathcal{M},q \rangle\rangle^+ \in PotArgsExt^+_{AF}(L:\langle \mathcal{N},k \rangle)} act(\Phi, \langle\langle \mathcal{M}, q \rangle\rangle^+) \wedge U(\Phi, \langle \mathcal{M}, q \rangle_{L:\langle \mathcal{N},k \rangle}) \qquad (\beta)$$

Making a similar analysis, but this time for the *undefeated* status of a node in a dialectical tree, we reach to the following sentence:

[1] Note that since $\langle \mathcal{N}, k \rangle$ is an argument w.r.t. $\mathcal{P} \cup \Phi$, by consistency condition of the definition of argument, we know that there is no strong information (in $\mathcal{P} \cup \Phi$) contradicting $\langle \mathcal{N}, k \rangle$, which in turn implies that there exist no empty argument $\langle \emptyset, q \rangle$ in $\mathcal{P} \cup \Phi$ contradicting $\langle \mathcal{N}, k \rangle$.

*$\langle \mathcal{N}, k\rangle_L$ will be defeated **iff** for all (non empty) activable potential argument $\langle\langle \mathcal{M}, q\rangle\rangle^+$ w.r.t. AF which extends $L : \langle \mathcal{N}, k\rangle$, it holds that **either** Φ does not activate $\langle\langle \mathcal{M}, q\rangle\rangle^+$ **or** Φ does activate $\langle\langle \mathcal{M}, q\rangle\rangle^+$ **and also** $\langle \mathcal{M}, q\rangle_{L:\langle \mathcal{N},k\rangle}$ is defeated.*

The next formula captures the meaning of the previous sentence:

$$U(\Phi, \langle \mathcal{N}, k\rangle_L) =_{def} \qquad (\gamma)$$
$$\bigwedge_{\langle\langle \mathcal{M},q\rangle\rangle^+ \in PotArgsExt^+_{AF}(L:\langle \mathcal{N},k\rangle)} \neg act(\Phi, \langle\langle \mathcal{M}, q\rangle\rangle^+) \vee \big(act(\Phi, \langle\langle \mathcal{M}, q\rangle\rangle^+) \wedge D(\Phi, \langle \mathcal{M}, q\rangle_{L:\langle \mathcal{N},k\rangle})\big)$$

Finally, we will present an example illustrating how the previous formulas can be used.

Example 5.1. *Consider an A-DeLP framework $AF_2 = \langle \Pi^2_f \cup \Delta_2, A_2\rangle$, where*
$\Pi^2_f = \{e, b\}$
$\Delta_2 = \{(a \prec b, c), (\sim a \prec d), (\sim a \prec g), (d \prec e), (g \prec c), (\sim d \prec e, f)\}$
$A_2 = \{c, \sim d, f, \sim g\}$
Next, we list all the (non empty) activable potential arguments w.r.t. AF_2:

$\langle\langle A_1, a\rangle\rangle^+$, *where* $A_1 = \{a \prec b, c\}$
$\langle\langle A_2, \sim a\rangle\rangle^+$, *where* $A_2 = \{(\sim a \prec d), (d \prec e)\}$
$\langle\langle A_3, \sim d\rangle\rangle^+$, *where* $A_3 = \{\sim d \prec e, f\}$
$\langle\langle A_4, \sim a\rangle\rangle^+$, *where* $A_4 = \{(\sim a \prec g), (g \prec c)\}$

Suppose we want to explain the warrant of a in AF_2. Then we need first to instantiate (and generate) the parametric formula (α) for AF_2 and taking h as a. The instantiated formula follows:

$$exp\_warrant(\Phi, a) = act(\Phi, \langle\langle A_1, a\rangle\rangle^+) \ \wedge \ U(\Phi, \langle A_1, a\rangle_{[\,]})$$

But the atom $U(\Phi, \langle A_1, a\rangle_{[\,]})$ in the previous instantiated formula denotes also an entire formula, that could be obtained instantiating (γ). The resulting instantiated formula follows next:

$$U(\Phi, \langle A_1, a\rangle_{[\,]}) = \neg act(\Phi, \langle\langle A_2, \sim a\rangle\rangle^+) \vee \big(act(\Phi, \langle\langle A_2, \sim a\rangle\rangle^+) \wedge D(\Phi, \langle A_2, \sim a\rangle_{[\langle A_1,a\rangle]})\big)$$
$$\wedge$$
$$\neg act(\Phi, \langle\langle A_4, \sim a\rangle\rangle^+) \vee \big(act(\Phi, \langle\langle A_4, \sim a\rangle\rangle^+) \wedge D(\Phi, \langle A_4, \sim a\rangle_{[\langle A_1,a\rangle]})\big)$$

Now two new atoms have been introduced, this time denoting formulas that can be obtained instantiating formula (β). We keep on doing that until no more atoms are introduced. The remaining instantiated formulae follows next:

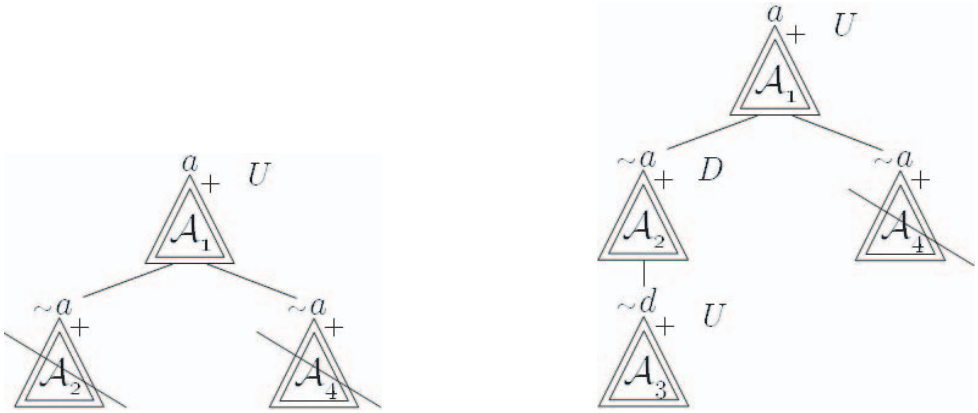
$D(\Phi, \langle A_2, \sim a\rangle_{[\langle A_1,a\rangle]}) = act(\Phi, \langle\langle A_3, \sim d\rangle\rangle^+) \ \wedge \ U(\Phi, \langle A_3, \sim d\rangle_{[\langle A_1,a\rangle,\langle A_2,\sim a\rangle]})$
$D(\Phi, \langle A_4, \sim a\rangle_{[\langle A_1,a\rangle]}) = false$
$U(\Phi, \langle A_3, \sim d\rangle_{[\langle A_1,a\rangle,\langle A_2,\sim a\rangle]}) = true$

Finally, starting form the instantiated formula for $exp\_warrant(\Phi, a)$ and substituting each D and U atom by its definition, we obtain the following formula, in which only act/1 atoms occur:

$$exp\_warrant(\Phi, a) = act(\Phi, \langle\langle A_1, a\rangle\rangle^+)$$
$$\wedge$$
$$\Big(\neg act(\Phi, \langle\langle A_2, \sim a\rangle\rangle^+) \vee \big(act(\Phi, \langle\langle A_2, \sim a\rangle\rangle^+) \wedge act(\Phi, \langle\langle A_3, \sim d\rangle\rangle^+)\big)\Big) \wedge \neg act(\Phi, \langle\langle A_4, \sim a\rangle\rangle^+)$$

That formula clearly specifies which activable potential arguments must Φ activate and which must not in order to be an explanation for the warrant of a. Let´s analyze what is concretely saying.

Note that the $\vee$ in the formula reflect alternative restrictions. In fact, the formula describes two alternative situations on which Φ constitutes an explanation for a. One is that Φ activates $\langle\langle A_1, a\rangle\rangle^+$ and does not activate $\langle\langle A_2, \sim a\rangle\rangle^+$ nor $\langle\langle A_4, \sim a\rangle\rangle^+$. The other is that Φ activates $\langle\langle A_1, a\rangle\rangle^+$, $\langle\langle A_2, \sim a\rangle\rangle^+$ and $\langle\langle A_3, \sim d\rangle\rangle^+$ and does not activate $\langle\langle A_4, \sim a\rangle\rangle^+$. The dialectical trees warranting a associated to each alternative are shown in figures 5a and 5b.



(a) (b)

Figure 5: Alternative dialectical trees warranting a.

6 CONCLUSIONS AND FUTURE WORK

In this article, we presented the confluence of two general ideas: Defeasible Logic Programming and Abduction in logic. Concretely, we introduced a framework that formally states the problem of doing abduction (obtaining abductive explanations) in the particular case that the abductive theory is a DeLP Program. In the reminder of this work, we faced that problem, providing a formal characterization of the notion of abductive explanation, from which we could then easily calculate the explanations. That characterization was presented in two parts. On the one hand, we formally stated the conditions that must satisfy an explanation in order to activate a given argument. On the other hand, we formally stated which arguments must an explanation activate and which must not in order to effectively explain the warrant of a literal.

As future work, we will develop a procedure for efficiently calculating the explanations. That procedure will be tightly based on the formal characterization presented here. Then as we have outlined in this article, by proving the correctness of the characterization, we cold ensure the correctness of the procedure.

REFERENCES

[1] Delp web page: http://lidia.cs.uns.edu.ar/delp.

[2] A. García and G. Simari. Defeasible logic programming: An argumentative approach. *Theory Practice of Logic Programming*, 4(1):95–138, 2004.

[3] Vladimir Lifschitz. Foundations of logic programs. In G. Brewka, editor, *Principles of Knowledge Representation*, pages 69–128. CSLI Pub., 1996.

Genetic Algorithms for Topical Web Search: A Study of Different Mutation Rates *

Rocío L. Cecchini[†], Carlos M. Lorenzetti[‡],
Ana G. Maguitman[‡] and Nélida Beatriz Brignole[†§]

{cr,cml,agm,nbb}@cs.uns.edu.ar

[†] LIDeCC - Laboratorio de Investigación y Desarrollo en Computación Científica
[‡] LIDIA - Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Av. Alem 1253, (8000) Bahía Blanca, Argentina
phone: 54-291-4595135 fax: 54-291-4595136

[§]Planta Piloto de Ingeniería Química (UNS-CONICET)
Cno la Carrindanga km 7, (8000) Bahía Blanca, Argentina

Abstract

Harvesting topical content is a process that can be done by formulating topic-relevant queries and submitting them to a search engine. The quality of the material collected through this process is highly dependant on the vocabulary used to generate the search queries. In this scenario, selecting good query terms can be seen as an optimization problem where the objective function to be optimized is based on the effectiveness of a query to retrieve relevant material. Three characteristics of this optimization problem are (1) the high-dimensionality of the search space, where candidate solutions are queries and each term corresponds to a different dimension, (2) the existence of acceptable suboptimal solutions, and (3) the possibility of finding multiple solutions. This paper describes optimization techniques based on Genetic Algorithms to evolve "good query terms" in the context of a given topic. We discuss the use of a mutation pool to allow the generation of queries with novel terms, and study the effect of different mutation rates on the exploration of query-space.

Keywords: topical web search, genetic algorithms, query formulation, diversity, query optimization

1 INTRODUCTION

Topical web portals have the purpose of gathering resources on specific subjects. The collected material is used to build specialized search and directory sites. Typically, focused crawlers are in charge of mining the Web to harvest topical content and populate the indices of these portals [6, 14]. As an alternative to focused crawlers, the process of harvesting topical content can be done by formulating topical queries to a search engine and selecting from the answer set those resources that are related to the topic at hand.

*This research work is supported by Agencia Nacional de Promoción Científica y Tecnológica (PICT 2005 Nro. 32373), Universidad Nacional del Sur (PGI 24/ZN13) and Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

To access topical information, appropriate queries must be formed. Finding good combinations of query terms requires exploring different direction of the space of potential queries. This exploration may require going beyond the initial set of terms by incorporating novel terms, which may prove to be useful at the moment of retrieving relevant material.

Refining search queries for topical search can be seen as an optimization problem, in which the search space is defined as the set of possible queries that can be presented to a search engine. The objective function to be optimized is based on the effectiveness of a query to retrieve relevant material when presented to a search engine. Depending on the system goals, a measure of query quality can be defined using traditional information retrieval notions such as precision and recall, or other customized performance evaluation metrics.

Solving this optimization problem is challenging because the query space has a huge number of dimensions, where each possible term accounts for a different dimension. However, for successful topical Web search high-quality queries are useful even if these queries are not the optimal ones. Therefore, it is usually sufficient to identify suboptimal solutions, a characteristic that helps alleviate our optimization problem. Another aspect of the optimization problem at hand is that we may be interested in finding many high-quality queries rather than a single one.

These characteristics make Genetic Algorithms (GAs) good candidates to tackle the problem of finding high-quality queries. This paper describes a framework based on GAs that addresses the problem of reflecting topical information when formulating search queries. The framework, discussed in detail in [5], takes an incremental approach to evolve high-quality queries for retrieving context-relevant textual resources (such as html pages, pdf files, Word files, etc.). It starts by generating an initial population of queries using terms extracted from a thematic context and incrementally evolves those queries based on their ability to retrieve relevant results when presented to a search engine. The contribution of this paper is a study of the effect that different mutation rates have on search results' diversity and quality.

In the next section we review the fundamental concepts of GAs. Then, in section 3 we overview our GA approach for evolving high-quality queries. Section 4 presents a study of the effect of different mutation rates (no mutation, classical mutation and hypermutation) on the diversity and quality of search results. The paper closes with a summary of our conclusions and a discussion of future work.

2 GENETIC ALGORITHMS

GAs [8] are robust optimization techniques based on the principle of natural selection and survival of the fittest, which claims "in each generation the stronger individual survives and the weaker dies". Therefore, each new generation would contain stronger (fitter) individuals in contrast to its ancestors.

To use GAs in optimization problems we need to define candidate solutions by chromosomes consisting of genes and a fitness function to be maximized. A population of candidate solutions (usually of a constant size) is maintained. The goal is to obtain better solutions after some generations. To produce a new generation GAs typically use selection together with the genetic operators of crossover and mutation. Parents are selected to produce offspring, favoring those parents with highest values of the fitness function. Crossover of population members takes place by exchanging subparts of the parent chromosomes (roughly mimicking a mating process), while mutation is the result of a random perturbation of the chromosome (e.g., replacing a gene by another). A simple GA works as follows:

Step 1: Start with a randomly generated population

Step 2: Evaluate the fitness of each individual in the population

Step 3: Select individuals to reproduce based on their fitness

Step 4: Apply crossover with probability Pc

Step 5: Apply mutation with probability Pm

Step 6: Replace the population by the new generation of individuals

Step 7: Go to step 2

Although selection, crossover and mutation can be implemented in many different ways, their fundamental purpose is to explore the search space of candidate solutions, improving the population at each generation by adding better offspring and removing inferior ones.

3 GENETIC ALGORITHMS FOR EXPLORING QUERY SPACE

The goal of our research work is to evolve queries that have the capability of retrieving topic-relevant content when presented to a search interface. In order to accomplish this goal we start with a population of queries composed of terms extracted from a topic description and rate each query according to the quality of the search results. As generations pass, queries associated with improved search results will predominate. Furthermore, the mating process continually combines these queries in new ways, generating ever more sophisticated solutions. In particular, the mutation mechanisms can be implemented in such a way that novel terms, i.e., terms that are not in the initial topic description, are brought into play.

3.1 Population and Representation of Chromosomes

The search space Q is constituted by all the possible queries that can be formulated to a search engine. Thus the population of chromosomes is a subset of such queries. Consequently, each chromosome is represented as a list of terms, where each term corresponds to a gene that can be manipulated by the genetic operators. The population is initialized with a fixed number of queries randomly generated with terms from the description of the topic at hand. The number of terms in each of the initial queries will be random, with a constant upper bound on the query size. While all terms in the initial population of queries come from the initial topic description, novel terms can be included in the queries after mutation takes place. These novel terms are obtained from a *mutation pool*, which is an ever increasing set of terms that may or may not be part of the topic description.

3.2 Fitness Function

We associate with the search space Q a fitness function Fitness : $Q \rightarrow [0 \ldots 1]$ which can numerically evaluate individual queries. The fitness function defines the criterion for assessing the quality of a query. Our conception of high-quality query is based on the query's ability to retrieve material similar to the topic of interest t when submitted to a search engine. The function we propose to measure fitness is

$$\text{Fitness}(\mathbf{q}) = \max_{d_i \in \mathbf{A_q}} (\sigma(t, d_i))$$

where $\mathbf{A_q}$ is the answer set for query $\mathbf{q}$ (set of documents returned by a search engine when $\mathbf{q}$ is used as a query) and $\sigma : D \times D \rightarrow [0 \ldots 1]$ is the similarity measure for a pair of documents (note that the topic t can be regarded as a document in D).

Different similarity measures, such as the standard cosine similarity or Jaccard similarity [1], can be used in the implementation of the fitness function. One pragmatic difficulty is the use of the complete answer set $\mathbf{A_q}$ in our definition of fitness. Looking at the entire set of pages returned by a search engine is too expensive for practical purposes. Therefore, we only look at the top ten results and only the "snippets" returned by the search engine are used for computing similarity. (The snippet is a text excerpt from the page summarizing the context where the search terms occur.)

3.3 Genetic Operators

A new generation in our GA is determined by a set of operators that select, recombine and mutate queries of the current population.

- **Selection:** A new population is generated by probabilistically selecting the highest-quality queries from the current set of queries. The probability that a query $\mathbf{q}$ will be selected is proportional to its own fitness and is inversely proportional to the fitness of the other queries in the current population. This method is known as the roulette-wheel selection.

- **Crossover:** Some of the selected queries are carried out into the next generations as they are, while others are recombined to create new queries. The recombination of a pair of parent queries into a pair of offspring queries is carried out by copying selected terms from each parent into the descendants. The crossover operator used in our proposal is known as single-point. It results in new queries in which the first n terms are contributed by one parent and the remaining terms by the second parent, where the crossover point n is chosen at random.

- **Mutation:** Small random changes can be produced to the new population of queries. These changes consist in replacing a randomly selected query term t^q by another term t^p. The term t^p is obtained from a *mutation pool* (described next).

3.4 Mutation Pool

The mutation pool is a set of terms that initially contains terms extracted from the description of the topic under analysis. As the system collects relevant content, the mutation pool is updated with new terms from the snippets returned by the search engine. This procedure brings new terms to the scene, allowing a broader exploration of the search space.

3.5 System Architecture

Figure 1 depicts the proposed system architecture for a topic-based search system based on GAs, which goal is to harvest resources for a topical portal. In the proposed prototype, the system maintains an internal representation of the topic at hand. In addition it maintains a population of queries which is incrementally refined as the system evolves. The basic mechanisms that enable the system to evolve queries and retrieve topic-relevant results are the following:

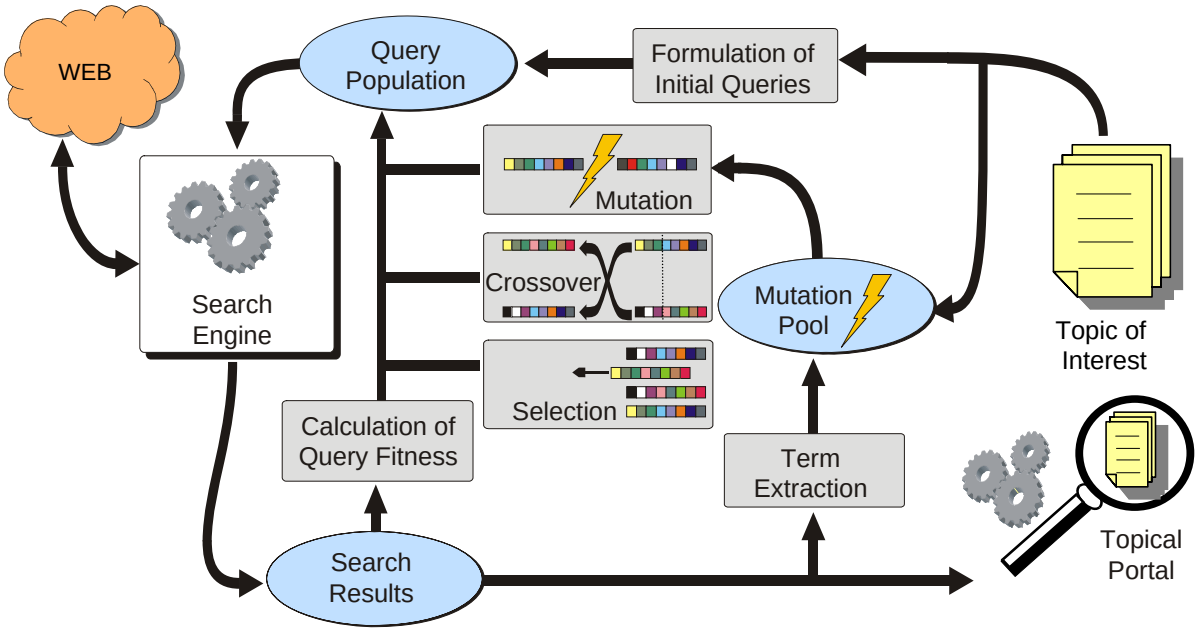


Figure 1: Architecture for a topical search system based on GAs. The collected results are used to populate the indices of a topical portal

- **Formulation of Initial Queries**. It selects terms from the topic of interest (consisting of a document or a set of related documents) and forms suitable queries, which are submitted to a standard search engine (e.g, Google) or entered into individual search forms (e.g., Amazon or PubMed). Initial queries will be automatically formed using a random selection of terms from the topic of interest. The sizes of the initial queries are never more than a predefined constant.

- **Calculation of Query Fitness**. This mechanism estimates the relevance of the results returned by a search engine after submitting a query. Based on the estimated relevance it will associate a fitness value with the query. One way the relevance of a search result can be approximated is by computing the similarity between the collected material and the topic of interest, but other approaches can be taken.

- **Term Extraction**. This component uses the content returned by a search engine to extract new terms, which are used to update the mutation pool.

- **Selection, Crossover and Mutation**. These mechanisms, described in section 3.3, are in charge of selecting, recombining and mutating the queries of the current population.

Although the sizes of the initial queries are never more than a predefined constant, the sizes of some queries in subsequent generations can exceed this limit. This is because applying the crossover operator can change the offspring size. Notice that existing search engines use up to a fixed number of terms and ignore subsequent ones (e.g., Google's query size limit is 32 terms). Interestingly, the eventual increase of query size beyond this limit captures, in a rough sense, the phenomenon of recessive inheritance: some terms that are ignored in a generation (because they occur beyond the query size limit) may be taken into account in subsequent generations when these terms become part of an offspring query after crossover takes place.

4 THE EFFECT OF DIFFERENT MUTATION RATES

In this section we analyze the effect of different mutation rates used by the implemented GAs for addressing our optimization problem. This analysis required access to different topic descriptions. We generated three topic descriptions using webpages from topics selected from the DMOZ directory (dmoz.org). The topics selected for our tests are *Business*, *Recreation* and *Society*. For each of our tests we run the GA five times. Each run consisted in 20 generations, with a population of 60 queries. The population of queries was randomly initialized using the corresponding topic description. The size of each query was a random number between 1 and 32. In all our tests the crossover probability was set to 0.7. To analyze the effect of different mutation rates we tested three different settings for the mutation probability: Pm=0 (no mutation), Pm=0.03 (classical mutation) and Pm=0.3 (hypermutation).

To evaluate the performance of topic-based retrieval based on GAs we adopted evaluation criteria based on the quality of the best queries at each generation. In order to propose a measure of query quality we first give a precise definition of similarity between a topic description and a retrieved result. Assume t is a topic description and $\mathbf{q}$ a query associated with t. Let $\mathbf{A_q} = \{a_1, \ldots, a_n\}$ be the set of retrieved resources (answer set) for $\mathbf{q}$. A measure of similarity between t and a_i can be computed using the *cosine similarity* defined as:

$$\sigma(t, a_i) = \frac{\overrightarrow{t} \cdot \overrightarrow{a}_i}{\| \overrightarrow{t} \| \cdot \| \overrightarrow{a}_i \|}$$

where $\overrightarrow{t}$ is the vector representation of the topic description based on the terms in t, and $\overrightarrow{a}_i$ is the vector representation of a_i based on the terms occurring in the corresponding snippet returned by a search engine.

We use σ to define *query quality based on maximum similarity* as follows:

$$\text{Quality\_Max}(\mathbf{q}) = \max_{a_i \in \mathbf{A_q}} (\sigma(t, a_i)).$$

Notice that the function Quality_Max is defined exactly as the fitness function presented in section 3.2.

4.1 The Effect of Mutation on Diversity

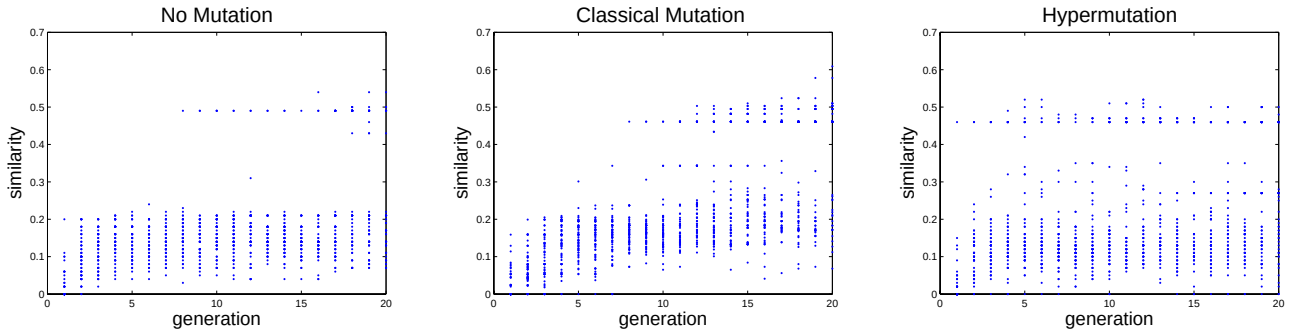


Figure 2: Scatter plots showing the distribution of similarity values for the best results associated with the individuals at each generation with Pm=0 (left), Pm=0.03 (center) and Pm =0.3 (right) for the topic *Business*.

In figure 2 we present three plots that allow us to visualize the evolution of a population of 60 queries across 20 generations for a description of the topic *Business* using different mutation rates. An interesting observation is that the higher the mutation rate, the earlier the algorithm starts to achieve higher similarity scores as well as more diversity. This is consistent with our intuitions, and highlights the importance of mutation at the moment of exploring the query space.

4.2 Evaluation of Query Quality

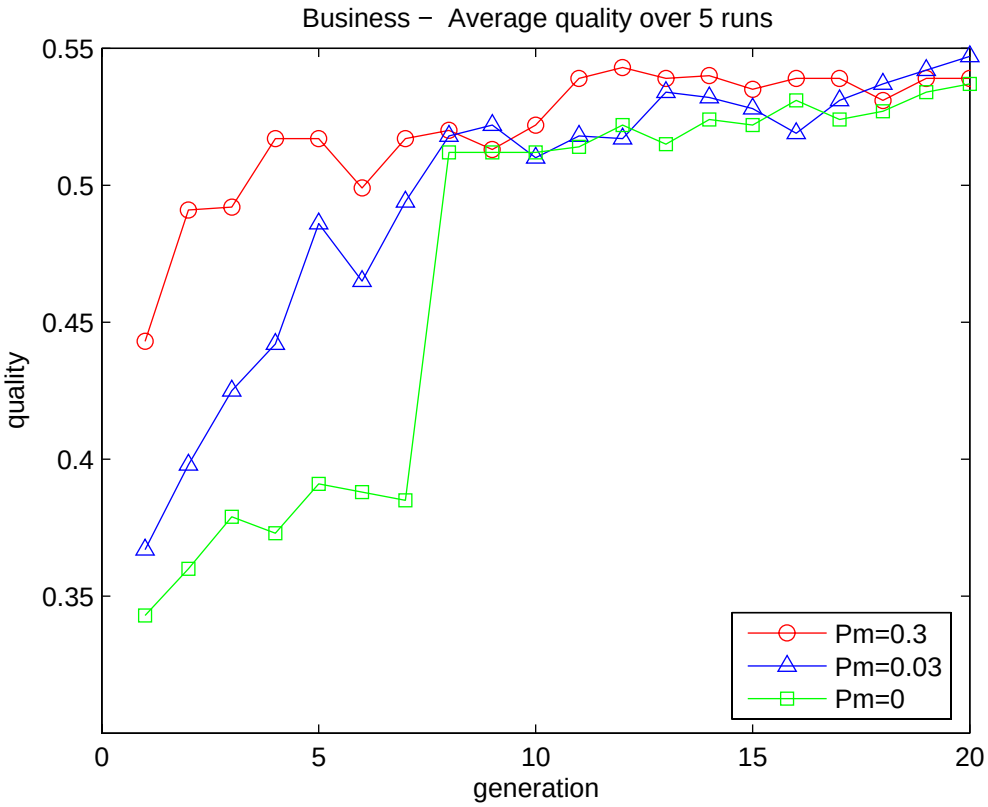


Figure 3: A test showing the average query quality over five independent runs for the topic *Business* using no mutation (Pm=0), classical mutation (Pm=0.03) and hypermutation (Pm=0.3).

Figures 3, 4 and 5 show the performance of the GA for the topics *Business*, *Recreation* and *Society*, respectively. For each topic we analyzed the effect of running the GAs without mutation, with classical mutation and with hypermutation. In these figures, we plotted the quality of the best query at each generation, averaged over the five runs. An interesting observation is that in all the tests, the case with Pm=0 (no mutation) results in the one with the slowest convergence rate towards high-quality queries.

Finally, we performed a statistical analysis to compare the query quality obtained at generation 1 with that obtained at generation 20. Tables 1, 2 and 3 show that for all tests performed there is an important improvement in query quality after 20 generations, and in most cases the improvement is statistically significant (C.I. highlighted in the tables). This allows us to conclude that, in general, the GA is able to evolve queries with quality considerably superior to that of the queries generated directly from the topic description.

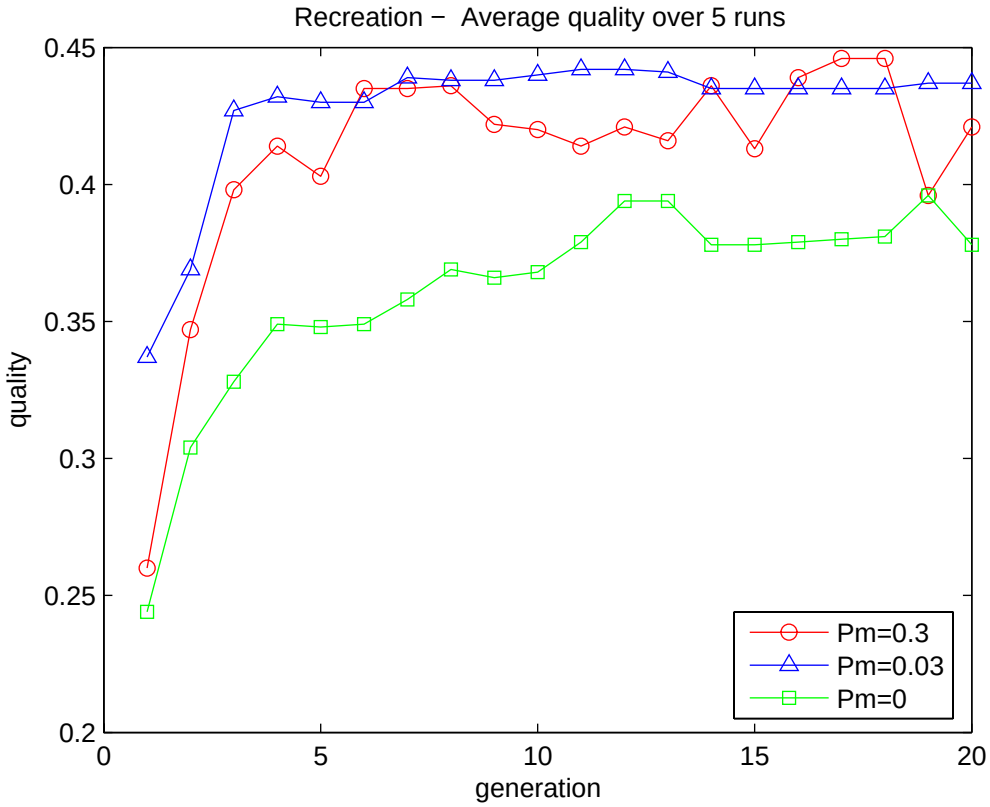


Figure 4: A test showing the average query quality over five independent runs for the topic *Recreation* using no mutation (Pm=0), classical mutation (Pm=0.03) and hypermutation (Pm=0.3).

	MEAN	95% C.I.
g=1	0.343	**(0.264,0.421)**
g=20	0.537	**(0.500,0.574)**

Pm=0

	MEAN	95% C.I
g=1	0.367	**(0.305,0.429)**
g=20	0.547	**(0.530,0.564)**

Pm=0.03

	MEAN	95% C.I.
g=1	0.443	(0.375,0.511)
g=20	0.539	(0.404,0.673)

Pm=0.3

Table 1: First Generation vs. Last Generation: confidence intervals for average query quality for topic *Business*.

	MEAN	95% C.I.
g=1	0.244	**(0.225,0.264)**
g=20	0.378	**(0.336,0.420)**

Pm=0

	MEAN	95% C.I.
g=1	0.337	**(0.289,0.385)**
g=20	0.437	**(0.395,0.479)**

Pm=0.03

	MEAN	95% C.I.
g=1	0.260	**(0.219,0.300)**
g=20	0.421	**(0.380,0.463)**

Pm=0.3

Table 2: First Generation vs. Last Generation: confidence intervals for average query quality for topic *Recreation*.

	MEAN	95% C.I.
g=1	0.220	**(0.202,0.237)**
g=20	0.313	**(0.243,0.383)**

Pm=0

	MEAN	95% C.I.
g=1	0.284	(0.258,0.311)
g=20	0.341	(0.304,0.378)

Pm=0.03

	MEAN	95% C.I.
g=1	0.235	(0.204,0.267)
g=20	0.302	(0.222,0.381)

Pm=0.3

Table 3: First Generation vs. Last Generation: confidence intervals for average query quality for topic *Society*.

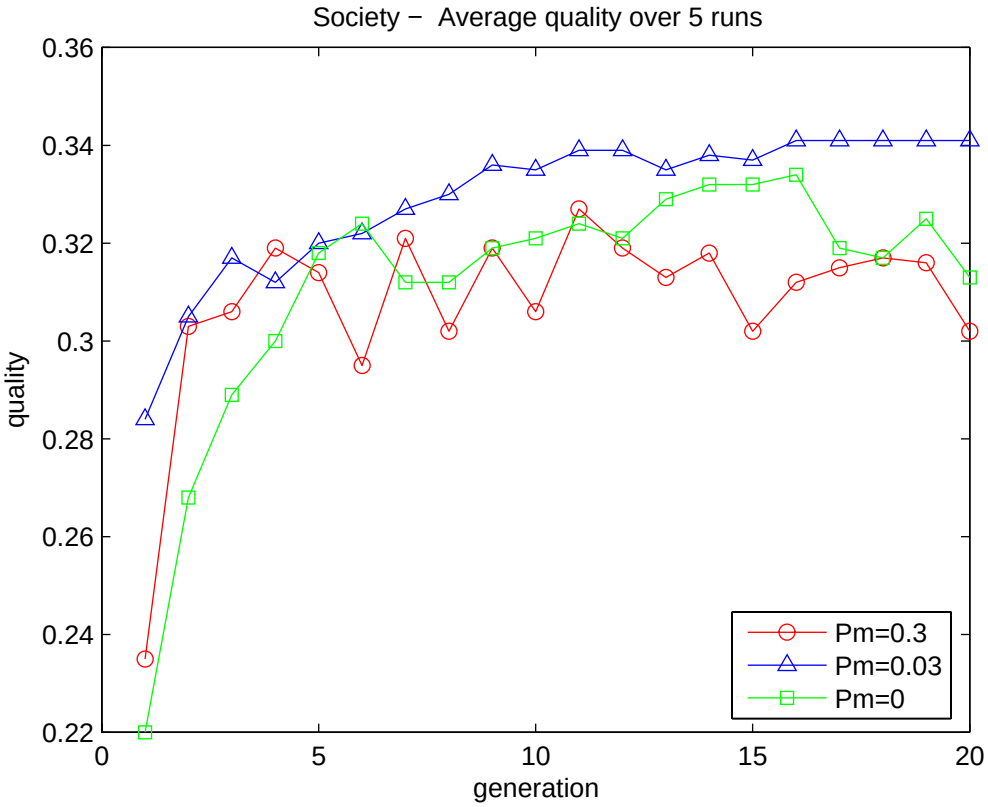


Figure 5: A test showing the average query quality over five independent runs for the topic *Society* using no mutation (Pm=0), classical mutation (Pm=0.03) and hypermutation (Pm=0.3).

5 CONCLUDING REMARKS

This paper presented an overview of techniques based on GAs for topical search. We have studied the effect of different mutation probabilities on the behavior of the proposed methods. As expected, we observed that higher mutations rates induce a more thorough exploration of the search space.

The techniques presented in this paper are applicable to any domain for which it is possible to generate term-based characterizations of a topic. Besides the application of resource harvesting for topical Web portals, these techniques can help build systems for a range of information services, including task-based search [11, 3], deep web search [9, 15] and support for knowledge management [10, 12]. However, we should remark that query adaptation involves submitting each new query to a search engine to calculate its fitness, which is a time consuming process. Therefore, we expect the proposed techniques to have potential applicability for non-real time systems, where slow response times are acceptable.

There have been some previous proposals to apply GA techniques to deal with problems in the area of information retrieval. Among the existing proposals we can mention the application of GA techniques to derive better document descriptions [7] and for term weight reinforcement in query optimization [18, 2]. These proposals differ from ours in attempting to tune the weights of individual terms rather than evolving queries. In addition, while our approach is fully automatic, others require relevance feedback from the users.

As part of our future work we expect to continue testing different settings for the GA parameters (population size, crossover probability, mutation probability) as well as other selection

methods such as tournament selection. We also plan to implement elitism, which will ensure the preservation of the best queries across generations. Other future directions include the application of genetic programming to evolve queries with special syntaxes [4] and the investigation of alternative fitness functions.

REFERENCES

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[2] Mohand Boughanem, Claude Chrisment, and Lynda Tamine. On Using Genetic Algorithms for Multimodal Relevance Optimization in Information Retrieval. *J. Am. Soc. Inf. Sci. Technol.*, 53(11):934–942, 2002.

[3] Jay Budzik, Kristian J. Hammond, and Larry Birnbaum. Information Access in Context. *Knowledge Based Systems*, 14(1–2):37–53, 2001.

[4] Tara Calishain and Rael Dornfest. *Google Hacks. 100 Industrial-Strengths Tips and Tools*. O'Reilly, 2003.

[5] Rocío L. Cecchini, Carlos M. Lorenzetti, Ana G. Maguitman, and Nélida B. Brignole. Searching the Web in Context: Genetic Algorithms for Exploring Query Space. To appear in *Proceedings of the Symposium of Information Society (SSI-JAIIO), Mar del Plata, Argentina*. SADIO, 2007.

[6] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused Crawling: a New Approach to Topic-Specific Web Resource Discovery. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1623–1640, 1999. 1999a.

[7] M. Gordon. Probabilistic and Genetic Algorithms in Document Retrieval. *Commun. ACM*, 31(10):1208–1218, 1988.

[8] John H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.

[9] Henry Kautz, Bart Selman, and Mehul Shah. The hidden Web. *AI Magazine*, 18(2):27–36, 1997.

[10] David Leake, Ana Maguitman, Thomas Reichherzer, Alberto Cañas, Marco Carvalho, Marco Arguedas, Sofia Brenes, and Tom Eskridge. Aiding Knowledge Capture by Searching for Extensions of Knowledge Models. In *Proceedings of KCAP-2003*. ACM Press, 2003.

[11] David B. Leake, Travis Bauer, Ana Maguitman, and David C. Wilson. Capture, Storage and Reuse of Lessons about Information Resources: Supporting Task-Based Information Search. In *Proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems. Austin, Texas*, pages 33–37. AAAI Press, 2000.

[12] Ana Maguitman, David Leake, and Thomas Reichherzer. Suggesting Novel but Related Topics: Towards Context-Based Support for Knowledge Model Extension. In *Proceedings of IUI-2005*, pages 207–214, New York, NY, USA, 2005. ACM Press.

[13] Ana Maguitman, David Leake, Thomas Reichherzer, and Filippo Menczer. Dynamic Extraction of Topic Descriptors and Discriminators: Towards Automatic Context-Based Topic Search. In *Proceedings of CIKM-2004*, Washington, DC, November 2004. ACM Press.

[14] Filippo Menczer, Gautam Pant, and Padmini Srinivasan. Topical Web Crawlers: Evaluating Adaptive Algorithms. *ACM Trans. Inter. Tech.*, 4(4):378–419, 2004.

[15] Alexandros Ntoulas, Petros Zerfos, and Junghoo Cho. Downloading Textual Hidden Web Content through Keyword Queries. In *Proceedings of JCDL-2005*, pages 100–109, New York, NY, USA, 2005. ACM Press.

[16] Bradley Rhodes and Thad Starner. The Remembrance Agent: A Continuously Running Automated Information Retrieval System. In *Proceedings of PAAM-1996*, pages 487–495, London, UK, April 1996.

[17] Barry Smyth and Paul McClave. Similarity vs. Diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning. Vancouver, Canada*, 2001.

[18] Jing-Jye Yang and Robert Korfhage. Query Optimization in Information Retrieval using Genetic Algorithms. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 603–613, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

An Argument-Based Approach to Cope with Trust and Pluralism in Web News Reports

Fernando M. Sagui[†§] **Ana G. Maguitman**[†§]
Carlos I. Chesñevar[†§] **Guillermo R. Simari**[†]
{fms,agm,cic,grs}@cs.uns.edu.ar
[†]Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Av. Alem 1253, (8000) Bahía Blanca, Argentina*
[§]CONICET - Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina

Abstract

Due to the huge amount of multi-source news that are available on the Web at any time, it is crucial to provide intelligent mechanisms to select and rank news reports. Over the last few years, a number of approaches based on criteria such as freshness, relevance and viewer profile have been proposed. However, most existing news processing services do not deal with credibility from a qualitative perspective, and do not provide mechanisms to cope with controversial news reports. To fill this gap, this paper proposes a news service framework that brings the notions of trust and pluralism into play. The proposed framework is based on a set of basic postulates characterizing the nature of trust. In our proposal, trust is modeled using Defeasible Logic Programming, a general-purpose defeasible argumentation formalism based on logic programming. Our approach helps identify antagonism among sources of news and facilitates the analysis of opposing positions. This allows us to integrate dialectical reasoning into a news recommender system, which has the capability of providing a reasoned basis for the news presented to the viewer.

Keywords: personalized news, Web, argumentation, trust, credibility, trustworthiness, pluralism

1 INTRODUCTION

According to several studies by Nielsen/NetRatings [5], reading news has become one of the most important activities on the Web. The number of visitors to news websites has steadily increased over the last years, and the abundant supply of online news is a clear indication of users' urge to be informed. Moreover, the availability of multiple sources of news provides an opportunity to access pluralistic opinions, which can be regularly found on the media.

Nowadays, there are several comercial multi-source news providers on the Web, such as Google News [3], Yahoo! News [4], MSNBC [2], etc. Although none of them has disclosed the technical details underlying the way news are selected, aggregated and ranked, it is evident that factors such as freshness, sources and popularity are taken into account. The information

*Partially supported by CONICET (PIP 5050), Agencia Nacional de Promoción Científica y Tecnológica (PICT 2005, Nro. 32373) and Universidad Nacional del Sur (24/N016 and 2/ZN13).

provided in news reports may not always be fully verifiable and therefore another important factor that can help select news is *trust* or *credibility*.

Research on multi-source news has generally overlooked the dynamics of news credibility, or if considered, it has been studied through quantitative approaches (e.g. [20]). There is also documented evidence [10] of Google News's plans to build a database of news source credibility based on information such as average story length, number of staff a news source employs, the volume of internet traffic to its website and the number of countries accessing the site. Google's approach to dealing with news credibility is to take all these and other parameters to create a single value used to rank the results of any news search. However, a foolproof approach to deal with news trust has not been developed yet and multi-source news services remain vulnerable to credibility breaches.[1]

A purely quantitative perspective to news credibility has several limitations. On the one hand, the absence of a formal model underlying quantitative approaches makes it hard to provide viewers with a justification of why certain news should be trusted. Because quantitative approaches are not equipped with inference capabilities, much of the implicit information remains undiscovered. On the other hand, they are incapable of dealing formally with the defeasible nature of trust. In addition, because trust is to a great degree subjective, quantifying trust by combining measures coming from a pool of credibility assessments may not be entirely realistic.

The goal of this research paper is to define a qualitative and personalized trust-based news service. The service will allow news viewers to access and compare the trustworthiness of news sources and their reports. Viewers' trust statements on sources and reports can be based on the viewers' subjective beliefs or, when absent, trust assumptions can be obtained indirectly from other viewers' beliefs. However, in order to derive trust from other viewers, a trust relationship between viewers must exist.

In this proposal, trust is modeled using DeLP, a defeasible argumentation framework based on logic programming [12]. This allows us to integrate dialectical reasoning into a news service, which will provide a reasoned basis for the news presented to the user.

The paper is organized as follows. The next section presents the background and motivations for our proposal, reviewing DeLP and discussing the problem of ranking and trusting Web news. Section 3 proposes a set of postulates for news trust and shows how to represent them by means of a set of DeLP rules. In section 4 we illustrate the proposal with a an example. Finally, section 5 overviews related work and section 6 outlines our conclusions.

2 BACKGROUND AND MOTIVATIONS

2.1 Defeasible logic programming

Defeasible logic programming (DeLP) [12] is a general-purpose defeasible argumentation formalism based on logic programming, intended to model inconsistent and potentially contradictory knowledge. A defeasible logic program has the form $\mathcal{P} = (\Pi, \Delta)$, where Π and Δ stand for *strict* and *defeasible* knowledge, respectively. The set Π involves *strict rules* of the form $P \leftarrow Q_1, \ldots, Q_k$ and *facts* (strict rules with empty body), and it is assumed to be *non-contradictory* (i.e., no complementary literals P and $\sim P$ can be inferred, where $\sim P$ denotes the contrary of P). The set Δ involves *defeasible rules* of the form $P \multimap Q_1, \ldots, Q_k$, which

[1] A famed example of this vulnerability problem is that of a fifteen-year old teenager who put out a fake release saying he had been hired by Google and within hours, his release was picked up in Google News.

stand for "$Q_1, \ldots Q_k$ *provide a tentative reason to believe* P." Rules in DeLP are defined in terms of *literals*. A literal is an atom A or the strict negation ($\sim A$) of an atom. Default negation (denoted *not* A) is also allowed in the body of defeasible rules (see [12] for details).

Deriving literals in DeLP results in the construction of *arguments*. An argument $\mathcal{A}$ for a literal Q (denoted $\langle \mathcal{A}, Q \rangle$) is a (possibly empty) set of ground defeasible rules that together with the set Π provide a proof for a given literal Q, satisfying the additional constraints of *non-contradiction* (*i.e.*, an argument should not allow the derivation of contradictory literals) and *minimality* (*i.e.*, the set of defeasible information used to derive Q should be minimal). Note that arguments are obtained by a mechanism similar to the usual query-driven SLD derivation from logic programming, performed by backward chaining on *both* strict and defeasible rules; in this context a negated literal $\sim P$ is treated just as a new predicate name *no_P*. In DeLP, arguments provide tentative support for claims (literals). Clearly, as a program $\mathcal{P}$ represents incomplete and tentative information, an argument $\langle \mathcal{A}, Q \rangle$ may be *attacked* by other arguments also derivable from $\mathcal{P}$. An argument $\langle \mathcal{B}, R \rangle$ is a *counter-argument* for $\langle \mathcal{A}, Q \rangle$ whenever a sub-argument $\langle \mathcal{A}', Q' \rangle$ (with $\mathcal{A}' \subseteq \mathcal{A}$) in $\langle \mathcal{A}, Q \rangle$ can be identified, such that $\langle \mathcal{B}, R \rangle$ and $\langle \mathcal{A}', Q' \rangle$ cannot be simultaneously accepted since their joint acceptance would allow contradictory conclusions to be inferred from $\Pi \cup \mathcal{A}' \cup \mathcal{B}$. If the attacking argument $\langle \mathcal{B}, R \rangle$ is preferred over $\langle \mathcal{A}', Q' \rangle$, then $\langle \mathcal{B}, R \rangle$ is called a *defeater* for $\langle \mathcal{A}, Q \rangle$. The preference criterion commonly used is *specificity* [12], preferring those arguments which are more direct or more informed, although other criteria could be adopted.

In DeLP the search for defeaters for a given argument $\langle \mathcal{A}, Q \rangle$ prompts a recursive process, resulting in the generation of a *dialectical tree*: the root node of this tree is the original argument at issue, and every children node in the tree is a defeater for its parent. Additional restrictions help to avoid circular situations when computing branches in a dialectical tree, guaranteeing that every dialectical tree is finite (see [12] for details). Nodes in the tree can be marked either as *defeated* (D-nodes) or as *undefeated* (U-nodes). The marking of the dialectical tree is performed as in an AND-OR trees: leaves are always marked as undefeated nodes (as they have no defeaters); inner nodes can be be marked either as undefeated (if and only if every of its children nodes is marked as defeated) or as defeated (whenever at least one of its children has been marked as undefeated). The original argument $\langle \mathcal{A}, Q \rangle$ (the root of tree) is deemed as ultimately acceptable or *warranted* whenever it turns out to be marked as undefeated after applying the above process.

Figure 1 shows an example of how DeLP can be used to represent commonsense knowledge about spiders. In this sample program there are four defeasible rules (e.g. spiders are usually dangerous, spiders which are dead are usually not dangerous, etc.). The program also includes some facts about a particular spider in a given situation (e.g. a black widow spider which looks dead, but moves when touched). By performing the query `dangerous(black_widow)`, DeLP allows us to conclude that we have a warranted argument supporting the claim that this spider is dangerous. Note that this involves computing a dialectical tree, which in this particular example involves just a single branch. As the query `dangerous(black_widow)` is supported by a warranted argument, the answer provided by DeLP is `yes`.

Note also that the computation of the dialectical tree is performed automatically by the DeLP interpreter on the basis of the program available. This process is based on an abstract machine which extends Warren's abstract machine for PROLOG [12]. Given a DeLP program $\mathcal{P}$, solving a query Q with respect to $\mathcal{P}$ may result in four possible answers: YES (there is at least one warranted argument $\mathcal{A}$ for Q); NO (there is at least one warranted argument $\mathcal{A}$ for $\sim Q$); UNDECIDED (none of the previous cases hold); and UNKNOWN (Q is not present in the

$\sim dead(X) \prec moves\_when\_touched(X)$
$dead(X) \prec looks\_dead(X)$
$\sim dead(X) \prec moves\_when\_touched(X),$
$looks\_dead(X)$
$\sim dangerous(X) \prec spider(X), dead(X)$
$dangerous(X) \prec spider(X)$
$spider(black\_widow)$
$looks\_dead(black\_widow)$
$moves\_when\_touched(black\_widow)$

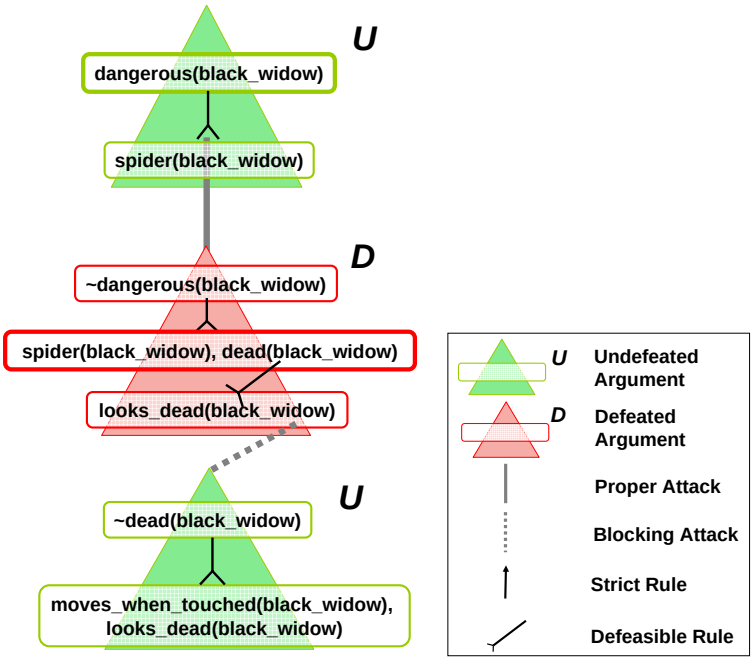


Figure 1: A sample DeLP program about spiders, and the dialectical analysis associated with solving the query `dangerous(black_widow)`

program signature). The emerging semantics is skeptical, computed by DeLP on the basis of the goal-directed construction and marking of dialectical trees, which is performed in a depth-first fashion. Additional facilities (such as visualization of dialectical trees, zoom-in/zoom-out view of arguments, etc.) are integrated in the DeLP environment to facilitate user interaction when solving queries.

2.2 Ranking and Trusting News

The problem of ranking Web news has attracted much attention in recent years. There are several reasons why measures of page authority such as PageRank [7] cannot be directly applied at the moment of ranking Web news. Differently from what happens with webpages, the Internet newspapers rarely use linking. Moreover, breaking news usually have priority over previous news because viewers prefer to see information about news events as soon as they take place. However, fresh news usually have very few incoming links, which precludes the application of link analysis algorithms to favor fresh news over stale ones. A ranking model that gives high priority to fresh news, however, will have some deficiencies. Reports on fresh news tend to be incomplete and many stories presented as breaking news are revised when additional information becomes available.

Another approach to rank news could be based on news popularity, estimated by monitoring the number of viewers accessing a report or by a system of voting on favorite stories. However, news popularity may not reflect the real value a news has for individual viewers.

Usually a viewer has to decide whether a news report is worth reading and whether the facts described in the report are credible. Unfortunately, developing an algorithm for Web news selection and ranking is very difficult because it needs to combine many, sometimes conflicting, aspects. The level of trust a viewer has on a piece of news is not necessarily associated with measures of news authority or popularity, and it may even be negatively correlated with news

freshness. A news service that uses a trust-management system can support the viewer in making the decision by selecting reports from trusted sources or based on another trustworthy viewer's opinion.

There are important aspects of trust that need to be considered in order to develop a realistic modeling of trust, as well as reliable and usable services based on this notion:

- **Trust should be Justified.** Most existing news ranking services act as "black boxes", because they refuse to disclose how they select certain news or rank them in certain ways. This results in trustworthiness issues because they do not provide viewers with a justification of why certain news should be trusted.

- **Trust is Defeasible.** News reports trusted by some viewer can be superseded by other reports carrying more authority, say from CNN or some other trusted source. In the meantime, news agencies are subject to time constraints, which results in the publication of reports with incomplete or inaccurate information. Trust on such reports could be revoked by the release of other more recent ones.

- **Trust is Subjective.** Like many aspects of the Web, news is becoming a collaborative activity. However, judgment of news credibility is idiosyncratic. Therefore, models that deal with trust as an objective notion are unrealistic. In this case, the opinion from the"wisdom of the crowds" may not be as useful as the viewer's personal opinion, or the opinion of another trusted viewer.

The rest of the paper is aimed at providing a framework for dealing with news credibility on the Web, in which the above characteristics of trust are taken into consideration.

3 A FORMAL FRAMEWORK FOR REASONING ABOUT NEWS TRUST

A system that reasons about news trust should take a number of ingredients into consideration:

- **Reports**. A report or news article is a written communication of a news event prepared by a specific news agency (source). When a report is made available on the Web, we can identify fields such as *title*, *source*, *timestamp*, *description*, *category* and *link to news content*. Other information related to the report such as *author* can also be derived in certain situations.

- **Sources**. The source of a news article is the agency in charge of supplying the report to be used by the media.

- **Viewers**. A viewer is a user of the news service. The system maintains a pool of viewers. Viewers can provide trust statements about reports, sources and other viewers.

- **Trust/Distrust Statements**. A trust (distrust) statement is an explicit assertion of the fact that a viewer trusts (distrusts) a report, a source or another viewer. These statements allow to infer implicit trust relations, which are useful to provide recommendations to the viewer based on trust.

3.1 Postulates for Trust Statements

This subsection presents a set of postulates for trust (and distrust) statements. The set of postulates will embody the general intuitions about the way trust and distrust statements could be derived from existing ones.

Postulate 1. A report coming from a trusted source will typically be trusted.

Postulate 2. A report coming from a distrusted source will typically be distrusted.

Postulate 3. A report trusted by a trusted viewer will typically be trusted.

Postulate 4. A report distrusted by a trusted viewer will typically be distrusted.

Postulate 5. A source trusted by a trusted viewer will typically be trusted.

Postulate 6. A source distrusted by a trusted viewer will typically be distrusted.

An interesting situation will arise when two conflicting conclusions can be reached. For example, a trusted viewer distrust a report, but the report was released by a trusted source. Assuming that in general we prefer to base our opinion on information provided by trusted sources, we can add the following two postulates:

Postulate 7. A report coming from a trusted source will typically be trusted, even if it is distrusted by a trusted viewer.

Postulate 8. A report coming from a distrusted source will typically be distrusted, even if it is trusted by a trusted viewer.

Additional postulates could be added. For example, if some viewer has a very good reputation for fact-checking, we will prefer to trust this viewer's opinion even if it conflicts with other viewers' opinions. Other postulates that could be added to the list may include references to the timestamp of the report. For instance, a more recent report will be trusted over an outdated one, unless the report is just out (due to the eagerness to publish the story some fresh news reports may not be as reliable as old ones). The list of postulates could be extended indefinitely, including references to news author, country of origin of the source, news category, etc. Some postulates could be personalized, because different viewers may disagree on the conclusion that should be adopted given certain facts. For the sake of simplicity, we will take postulates 1 to 8 as the core postulates for our trust-management system.

3.2 Using DeLP to Reason about News Trust and Pluralism

Postulates 1 to 8 can be naturally modeled using the following DeLP rules:

$$
\begin{array}{rcll}
trust\_report(V,R) & -\!\prec & report\_source(R,S), trust\_source(V,S) & (R_1)\\
\sim trust\_report(V,R) & -\!\prec & report\_source(R,S), \sim trust\_source(V,S) & (R_2)\\
trust\_report(V,R) & -\!\prec & trust\_viewer(V,V1), trust\_report(V1,R) & (R_3)\\
\sim trust\_report(V,R) & -\!\prec & trust\_viewer(V,V1), \sim trust\_report(V1,R) & (R_4)\\
trust\_source(V,S) & -\!\prec & trust\_viewer(V,V1), trust\_source(V1,S) & (R_5)\\
\sim trust\_source(V,S) & -\!\prec & trust\_viewer(V,V1), \sim trust\_source(V1,S) & (R_6)\\
trust\_report(V,R) & -\!\prec & report\_source(R,S), trust\_source(V,S), & \\
 & & trust\_viewer(V,V1), \sim trust\_report(V1,R) & (R_7)\\
\sim trust\_report(V,R) & -\!\prec & report\_source(R,S), \sim trust\_source(V,S), & \\
 & & trust\_viewer(V,V1), trust\_report(V1,R) & (R_8)
\end{array}
$$

We envision a trust-management system with built-in rules R_1 to R_8. We should remark that viewers will not need to deal directly with DeLP rules. However, through a user-friendly question-answering interface it will be possible to extend or adjust the built-in core rules based on the viewer's preferences. Trust and distrust statements about reports, sources and other viewers will be added to the system whenever the viewer rates these entities.

For a particular viewer v, and based on the corresponding DeLP rules and facts, news reports will be classified into three sets:

Trusted Reports: those reports r_i for which there exists at least one warranted argument supporting $trust\_report(v, r_i)$.

Distrusted Reports: those reports r_i such that there is a warranted argument supporting $\sim trust\_report(v, r_i)$.

Undecided: those reports r_i for which there is no warranted argument for $trust\_report(v, r_i)$ or $\sim trust\_report(v, r_i)$.

This classification will allow the viewer to focus on those reports considered trustworthy, and to be warned about the non trustworthy ones.

4 A WORKED EXAMPLE

Assume Joe is a viewer, whose personalized trust-management system contains rules R_1 to R_8 together with the following facts:

$$
\begin{array}{ll}
report\_source(true\_news, the\_truthteller) & (F_1) \\
report\_source(false\_news, the\_corker) & (F_2) \\
report\_source(some\_news, the\_incog) & (F_3) \\
trust\_source(joe, the\_truthteller) & (F_4) \\
\sim trust\_source(tom, the\_corker) & (F_5) \\
\sim trust\_report(ann, true\_news) & (F_6) \\
trust\_report(ann, false\_news) & (F_7) \\
\sim trust\_report(ann, some\_news) & (F_8) \\
trust\_report(tom, some\_news) & (F_9) \\
trust\_viewer(joe, ann) & (F_{10}) \\
trust\_viewer(joe, tom) & (F_{11})
\end{array}
$$

Suppose that the reports "True News", "False News" and "Some News" need to be classified based on their trustworthiness status. Figure 2 shows that "True News" can be trusted by Joe, as there exists a warranted argument supporting $trust\_report(joe, true\_news)$. On the other hand, the existence of a warranted argument for $\sim trust\_report(joe, false\_news)$, as shown in figure 3, allows the system to conclude that "False News" must be distrusted. Finally, figure 4 shows that "Some News" can neither be trusted nor distrusted as there is no warranted argument for $trust\_report(joe, some\_news)$ or $\sim trust\_report(joe, some\_news)$.

An important by-product of using an argument-based approach to classify news reports according to their trustworthiness status is that the viewer will be able to inspect the reasons that lead the system to provide a conclusion.

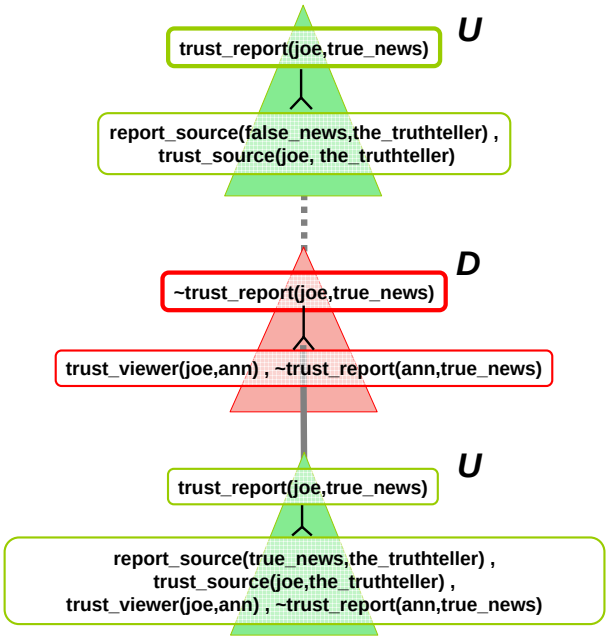


Figure 2: A DeLP dialectical tree supporting the conclusion that the "True News" report should be trusted by Joe.

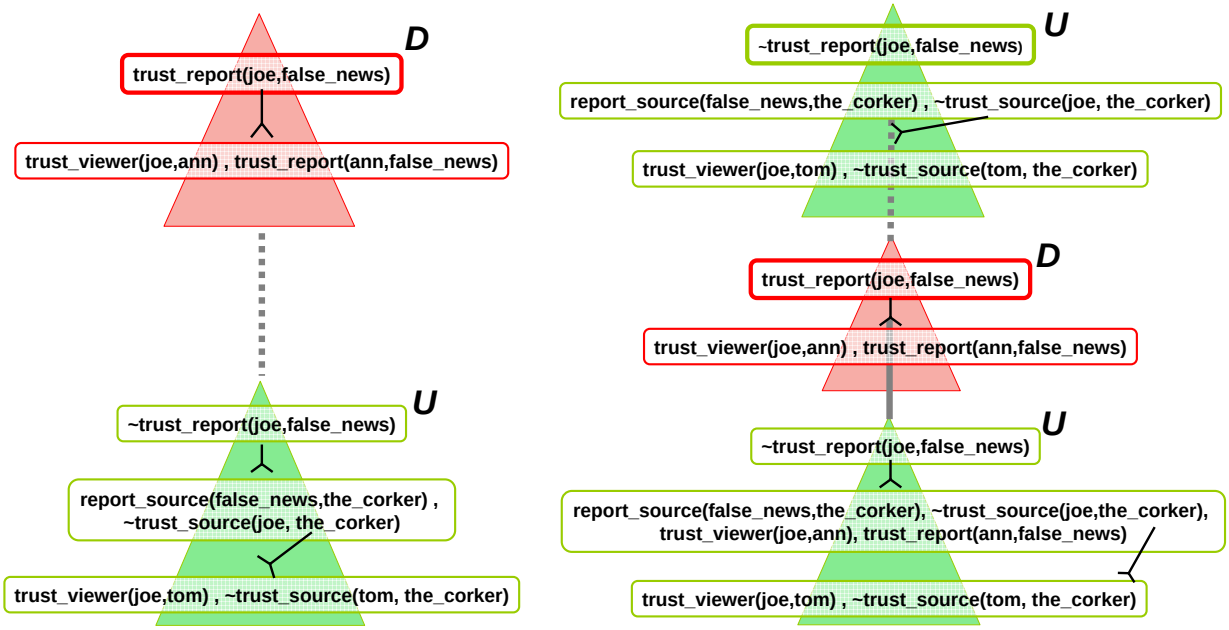


Figure 3: DeLP dialectical trees supporting the conclusion that the "False News" report should be distrusted by Joe.

5 RELATED WORK

A variety of methods have been proposed and a number of systems have been developed to facilitate access to news on the Web. NewsInEssence [23] is a system that searches and clusters related news. QCS [9] is a tool that facilitates the task of grouping and categorizing news. In [15] a method is proposed to search web articles while TV news are on the air. Other tools

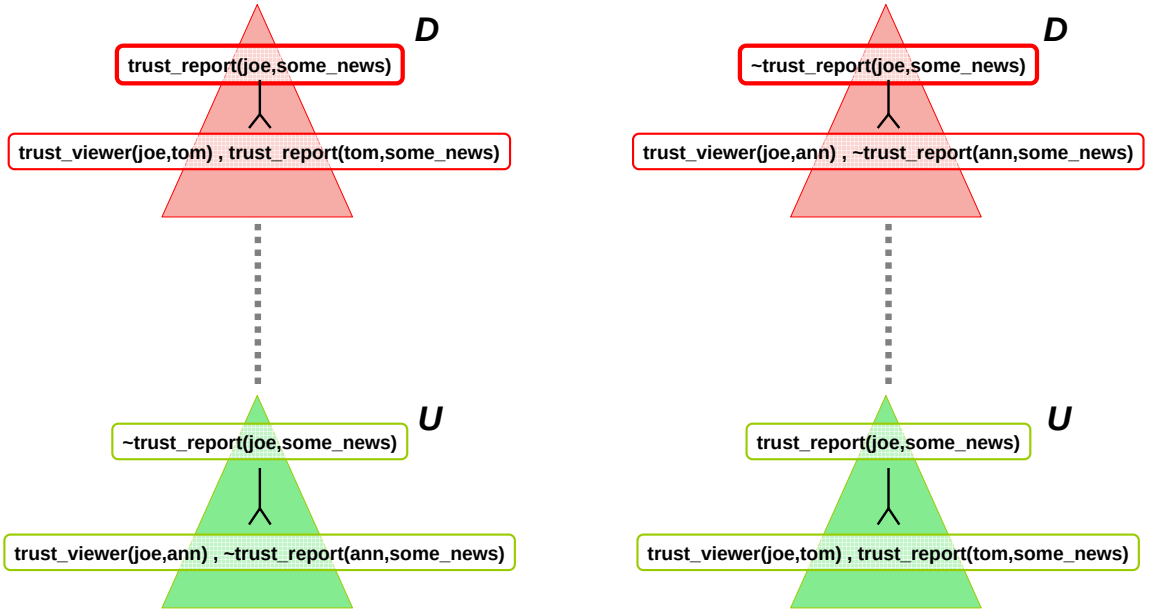


Figure 4: DeLP dialectical trees showing that it is not possible to conclude that the "Some News" report should be trusted or distrusted by Joe and therefore it will be classified as undecided.

(e.g. [24]) automatically extract domain-oriented news from websites. Velthune [14] is a search engine that extracts information both from the Web and from newsfeeds. Another system, NewsJunkie [11], has the purpose of identifying novel news in the context of stories previously reviewed by the user. Compare&Contrast [18] is a Case-Based Reasoning system that uses the Web as a knowledge base to discover comparable cases for news stories. Many techniques have been proposed to organize news in topics. Most of them try to recognize, track and summarize stories [21, 19, 17, 6]. ArgueNet, a system previously proposed by some of the authors [8], is an argument-based framework for ranking web results. The ArgueNet system differs from this proposal in being a general approach for identifying relevant results and not attempting to provide an underlying formal model of the notion of trust.

A few systems deal with the notions of credibility and media bias. In [20] a method is proposed to rate the credibility of news documents. The method uses algorithms that compare the content of different news sources. PolyNews [22] is a news service framework that tries to mitigate the effect of media bias by the creation of multiple classified viewpoints. NewsTrust [1] is a service created to evaluate news where users can rank news reports, news writers and news sources. A multi-layer recommendation system based on trust is proposed in [16] and a general framework for the analysis of the propagation of trust and distrust is presented in [13].

6 CONCLUSIONS

In this paper we have proposed a formal framework to deal with news trust and distrust. A set of postulates has been proposed and modeled using DeLP rules.

An implementation of the proposed trust-based system is in progress. As a first step we have implemented a multi-source news service that monitors several newsfeeds and populates a database of news (`http://cs.uns.edu.ar/f̃ms/newsdb/`). We are currently devel-

oping algorithms for clustering news reports based on the news event the reports describe. In the near future, we expect to integrate this environment with information about viewers and their trust statements, as well as to equip the system with reasoning capabilities based on DeLP as described in this article. The DeLP environment is available online to test at `http://lidia.cs.uns.edu.ar/delp_client`.

As part of our future work we expect to study more powerful models of trust. A user is much more likely to believe statements from a trusted acquaintance than from a stranger. Therefore, trust could be quantified and trust values could be transitively obtained from other viewer and scaled down depending on the viewer providing the information. In this sense we expect to study ways of extending our model of trust by adding possibilistic and probabilistic reasoning.

We believe our proposal presents a number of advantages over existing news recommendation services. An argument-based approach allows the exploration of arguments that challenge the viewer perspective. Using an argument-based framework can help the viewer review previously maintained trust statement, as it is possible to analyze the justification in favor and against trusting some report. This could also help discover events in which biased sources present facts in a convincing manner, but neglect to include other important facts that are against their position. Because conclusions are justified by the system, the viewer is able to go back and figure out where errors were revealed. Clearly, in practice most of such facts will remain unverifiable, and there will be room for ideology and interpretation. The proposed framework intends to be faithful to this aspect of reality.

REFERENCES

[1] http://beta.newstrust.net/.

[2] http://newsbot.msnbc.msn.com/.

[3] http://news.google.com/.

[4] http://news.yahoo.com/.

[5] http://www.nielsen-netratings.com/.

[6] James Allan, Rahul Gupta, and Vikas Khandelwal. Temporal summaries of new topics. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–18, New York, NY, USA, 2001. ACM Press.

[7] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[8] Carlos Chesñevar and Ana Maguitman. ArgueNet: An argument-based recommender system for solving Web search queries. In *Proceedings of the International IEEE Conference on Intelligent Systems (IS 2004)*, pages 282–287. IEEE, June 2004.

[9] Daniel M. Dunlavy, John Conroy, and Dianne P. O'Leary. Qcs: a tool for querying, clustering, and summarizing documents. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 11–12, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[10] Barry Fox. Google searches for quality not quantity. *New Scientist magazine*, 2497:24. 30 April 2005.

[11] Evgeniy Gabrilovich, Susan Dumais, and Eric Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 482–490, New York, NY, USA, 2004. ACM Press.

[12] A. García and G. Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.

[13] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *International World Wide Web Conference (WWW2004)*, 2004.

[14] A. Gulli. The anatomy of a news search engine. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 880–881, New York, NY, USA, 2005. ACM Press.

[15] Monika Henzinger, Bay-Wei Chang, Brian Milch, and Sergey Brin. Query-free news search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 1–10, New York, NY, USA, 2003. ACM Press.

[16] Claudia Hess. Trust-based recommendations for publications: A multi-layer network approach. *TCDL Bulletin*, 2(2), 2006.

[17] Andrew J. Kurtz and Javed Mostafa. Topic detection and interest tracking in a dynamic online news source. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 122–124, Washington, DC, USA, 2003. IEEE Computer Society.

[18] Jiahui Liu, Earl Wagner, and Larry Birnbaum. Compare&contrast: using the web to discover comparable cases for news stories. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 541–550, New York, NY, USA, 2007. ACM Press.

[19] Masaki Mori, Takao Miura, and Isamu Shioya. Topic detection and tracking for news web pages. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 338–342, Washington, DC, USA, 2006. IEEE Computer Society.

[20] Ryosuke Nagura, Yohei Seki, Noriko Kando, and Masaki Aono. A method of rating the credibility of news documents on the web. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 683–684, New York, NY, USA, 2006. ACM Press.

[21] Ramesh Nallapati, Ao Feng, Fuchun Peng, and James Allan. Event threading within news topics. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 446–453, New York, NY, USA, 2004. ACM Press.

[22] Souneil Park, Seungwoo Kang, and Junehwa Song. Polynews: Delivering multiple aspects of news to mitigate media bias. Technical report, KAIST, 2006.

[23] Dragomir Radev, Jahna Otterbacher, Adam Winkel, and Sasha Blair-Goldensohn. Newsinessence: summarizing online news topics. *Commun. ACM*, 48(10):95–98, 2005.

[24] D. C. Reis, P. B. Golgher, A. S. Silva, and A. F. Laender. Automatic web news extraction using tree edit distance. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 502–511, New York, NY, USA, 2004. ACM Press.

Intelligent Methods for Information Access in Context: The Role of Topic Descriptors and Discriminators

Carlos M. Lorenzetti[†] **Rocío L. Cecchini**[‡] **Ana G. Maguitman**[†]

e-mail: {cml,rlc,agm}@cs.uns.edu.ar
[†]Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
[‡]Laboratorio de Investigación y Desarrollo en Computación Científica
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Av. Alem 1253, (8000) Bahía Blanca, Argentina.*

Abstract

Successful access to information sources on the Web depends on effective methods for identifying the needs of a user and making relevant information resources available when needed. This paper formulates a theoretical framework for the study of context-driven Web search and proposes new methods for learning query terms based on the user task. These methods use an incrementally-retrieved, topic-dependent selection of Web documents for term-weight reinforcement reflecting the aptness of the terms in describing and discriminating the topic of the user context. Based on this framework, we propose an incremental search algorithm for information retrieval agents that has the potential to improve significantly over the traditional IR techniques. The new algorithm learns new descriptors by searching for terms that tend to occur *often* in relevant documents, and learns good discriminators by identifying terms that tend to occur *only* in the context of the given topic. We discuss the technical challenges posed by this new framework, outline our agent system architecture, and present an evaluation of the proposed techniques.

Keywords: context modeling, information retrieval, descriptors, discriminators

1 INTRODUCTION

The World Wide Web is an ever-expanding source of information about a huge diversity of topics. As a consequence, it is becoming increasingly important to know how to find out about a topic of special interest, focusing the search on material that is relevant to the current task. This search activity could be done more effectively if intelligent agents for information access and delivery were included as part of the system search tools. In order to reduce the user cognitive overload, task-specific Web search agents need to be adapted to deliver few but highly relevant resources.

An important requirement for these agents is to provide relevant material, doing it at the right time, and without causing undue or excessive distraction. Two elements that can be exploited by an agent to enhance Web search are *user context* and *user preferences*. User context reflects the task in which the user is immersed (e.g., [8, 14]). The context may consist of an electronic document the user is editing, Web pages the user has recently visited, etc. User preferences reflect the way in which a

*Partially supported by Agencia Nacional de Promoción Científica y Tecnológica (PICT 2005, Nro. 32373), Universidad Nacional del Sur (24/ZN13) and CONICET.

user would prioritize search results. User preferences could be entered explicitly by the user or could be inferred by the agent system (e.g., by monitoring the user's behavior).

As part of our research work we are studying how to build intelligent agents that can provide context-based and preference-based support by retrieving useful information from the Web. These agents monitor the user and search the Web for material related to the user current task and preferences. A general discussion of the proposed architecture and project goals can be found in [9].

This paper presents general techniques for incrementally learning important terms in the context of a user task. Specifically, we are studying three questions: (1) can the user context be usefully exploited by information retrieval agents to access relevant material on the Web?, (2) can a set of context-specific terms be incrementally refined, based on the analysis of search results?, and (3) are the context-specific terms learned by incremental methods better query terms than those identified by classical IR techniques? To address these questions section 2 presents a theoretical framework for the study of contextual search on the Web. Section 3 describes our agent system architecture for context-based Web retrieval. Section 4 shows the results of our evaluations. Section 5 discusses related work, and finally, section 6 presents our conclusions.

2 A FRAMEWORK FOR LEARNING CONTEXT-SPECIFIC TERMS

Search interfaces provide access to a vast repository of information on the World Wide Web. However, finding relevant information remains challenging, because of the need to select useful resources from an enormous range of possibilities. For many computer-mediated tasks, the user context provides a rich set of terms that can be exploited by intelligent information agents to enhance Web search. Such agents can be equipped with special monitoring capabilities, designed to generate a model of the user task. The agents will be in charge of observing how the user interacts with different kinds of computer utilities (such as email systems, browsers and text editors) to characterize the user's information needs as a collection of weighted terms. This requires a framework for learning context-specific terms.

Classical IR weighting schemes, such as *term frequency inverse document frequency* (TF-IDF) [23], are reasonable measures of term importance but are insufficient for the task domain for our research. As has been discussed by a number of sources, issues arise when attempting to apply conventional IR schemes for measuring term importance to systems for searching Web data [13, 5]. One difficulty is that methods for Web search do not have access to a fully predefined collection of documents, raising questions about the suitability of classical IR schemes for measuring term importance when searching the Web.

A central question addressed in our work is how to learn context-specific terms based on the user current task and an open collection of incrementally retrieved Web documents. In what follows, we will assume that a user task is represented as a set of cohesive terms summarizing the topic of the user context. Consider for example a topic involving the *Java Virtual Machine*, described by the following set of terms:

java	virtual	machine	programming	language
computers	netbeans	applets	ruby	code
sun	technology	source	jvm	jdk

Context-specific terms may play different roles. For example, the term *java* is a good descriptor of the topic for a general audience. On the other hand, terms such as *jvm* and *jdk*—which stand for "Java Virtual Machine" and "Java Development Kit"—may not be good descriptors of the topic for that audience, but are effective in bringing information similar to the topic when presented in a query. Therefore, *jvm* and *jdk* are good discriminators of that topic.

In previous work [18] we have tested the following two hypotheses:

- Good topic descriptors can be found by looking for terms that occur often in documents similar to the given topic.
- Good topic discriminators can be found by looking for terms that occur only in documents similar to the given topic.

Thus a possible strategy that an information agent can follow for finding good topic descriptors is to (1) find documents that are similar to other documents already known to include that topic, and (2) select from those documents the terms that occur often. On the other hand, a term is a good discriminator for a topic if most documents that contain that term are topically related. Therefore, finding good topic discriminators requires finding terms that tend to occur only in the context of the given topic. Both topic descriptors and discriminators are important as query terms. Because topic descriptors occur often in relevant pages, using them as query terms may improve recall. Similarly, good topic discriminators occur primarily in relevant pages, and therefore using them as query terms may improve precision.

As a first approximation to compute descriptive and discriminating power, we begin with a collection of m documents and n terms. As a starting point we build an $m \times n$ matrix $\mathbf{H}$, such that $\mathbf{H}[i,j] = k$, where k is the number of occurrences of term t_j in document d_i. In particular we can assume that one of the documents (e.g., d_0) corresponds to the initial user context. The following example illustrates this situation:

$$\mathbf{H} = \begin{array}{r} \\ \texttt{java} \\ \texttt{machine} \\ \texttt{virtual} \\ \texttt{language} \\ \texttt{programming} \\ \texttt{coffee} \\ \texttt{island} \\ \texttt{province} \\ \texttt{jvm} \\ \texttt{jdk} \end{array} \begin{array}{c} \begin{array}{ccccc} d_0 & d_1 & d_2 & d_3 & d_4 \end{array} \\ \begin{pmatrix} 4 & 2 & 5 & 5 & 2 \\ 2 & 6 & 3 & 2 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 1 & 1 \\ 3 & 0 & 2 & 2 & 0 \\ 0 & 3 & 0 & 0 & 3 \\ 0 & 4 & 0 & 0 & 2 \\ 0 & 4 & 0 & 0 & 1 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 3 & 3 & 0 \end{pmatrix} \end{array}$$

Documents:
d_0: user context
d_1: espressotec.com
d_2: netbeans.org
d_3: sun.com
d_4: wikitravel.org

The matrix $\mathbf{H}$ allows us to formalize the notions of good descriptors and good discriminators. We define *descriptive power of a term in a document* as a function $\lambda : \{d_0, \ldots, d_{m-1}\} \times \{t_0, \ldots, t_{n-1}\} \rightarrow [0,1]$:

$$\lambda(d_i, t_j) = \frac{\mathbf{H}[i,j]}{\sqrt{\sum_{k=0}^{n-1} (\mathbf{H}[i,k])^2}}.$$

If we adopt $\mathrm{s}(k) = 1$ whenever $k > 0$ and $\mathrm{s}(k) = 0$ otherwise, we can define the *discriminating power of a term in a document* as a function $\delta : \{t_0, \ldots, t_{n-1}\} \times \{d_0, \ldots, d_{m-1}\} \rightarrow [0,1]$:

$$\delta(t_i, d_j) = \frac{\mathrm{s}(\mathbf{H}[j,i])}{\sqrt{\sum_{k=0}^{m-1} \mathrm{s}(\mathbf{H}[k,i])}}.$$

Note that λ and δ satisfy the conditions

$$\sum_j (\lambda(d_i, t_j))^2 = 1 \quad \text{and} \quad \sum_j (\delta(t_i, d_j))^2 = 1.$$

Given a term t_i in a document d_j, the term t_i will have a high descriptive power in d_j if it occurs often in d_j, while it will have a high discriminating power if it tends to occur only in d_j (i.e., it seldom

occurs in other documents). The descriptive power and discriminating power values for the terms in the example given above are as follows:

$$\begin{array}{r} \\ \texttt{java} \\ \texttt{machine} \\ \texttt{virtual} \\ \texttt{language} \\ \texttt{programming} \\ \texttt{coffee} \\ \texttt{island} \\ \texttt{province} \\ \texttt{jvm} \\ \texttt{jdk} \end{array} \begin{array}{c} \lambda(d_0,t_j)^T \\ \left(\begin{array}{c} 0.718 \\ 0.359 \\ 0.180 \\ 0.180 \\ 0.539 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \end{array}\right. \end{array} \begin{array}{c} \delta(t_i,d_0) \\ \left.\begin{array}{c} 0.447 \\ 0.500 \\ 0.577 \\ 0.500 \\ 0.577 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \end{array}\right) \end{array}$$

The above weights reflect some of the limitations of this first approach. For instance, the weights associated with the terms *jvm* and *jdk* do not reflect their importance as discriminators of the topic under analysis. In the same way as the well-known TF and IDF measures [23], the functions λ and δ allow to discover terms that are good descriptors and good discriminators of a document, as opposed to good descriptors and good discriminators of the *topic* of a document.

Our current goal is to formulate notions of topic descriptors and discriminators suitable for the Web scenario. Rather than extracting descriptors and discriminators directly from the user context, we want to extract them from *the topic* of the user context. This requires an incremental method to characterize the topic of the user context, which is done by identifying documents that are similar to the user current context. Assume the user context and the retrieved Web documents are represented as document vectors in term space. To determine how similar two documents d_i and d_j are we adopt the IR cosine similarity [2]. This measure is defined as follows:

$$\sigma(d_i,d_j) = \sum_{k=0}^{n-1}[\lambda(d_i,t_k)\cdot\lambda(d_j,t_k)].$$

The similarity values between the user context (d_0) and the other documents in our example are as follows:

$$\sigma(d_0,d_j) = \begin{array}{cccc} d_1 & d_2 & d_3 & d_4 \\ \big(0.399 & 0.840 & 0.857 & 0.371\big) \end{array}$$

The notion of topic descriptors was informally defined earlier "as terms that occur *often* in the context of a topic." We define the *term descriptive power in the topic of a document* as a function $\Lambda : \{d_0,\ldots,d_{m-1}\} \times \{t_0,\ldots,t_{n-1}\} \rightarrow [0,1]$. If $\sum_{\substack{k=0\\k\neq i}}^{m-1}\sigma(d_i,d_k) = 0$ then we set $\Lambda(d_i,t_j) = 0$. Otherwise we define $\Lambda(d_i,t_j)$ as follows:

$$\Lambda(d_i,t_j) = \frac{\sum_{\substack{k=0\\k\neq i}}^{m-1}[\sigma(d_i,d_k)\cdot[\lambda(d_k,t_j)]^2]}{\sum_{\substack{k=0\\k\neq i}}^{m-1}\sigma(d_i,d_k)}$$

Thus, the descriptive power of a term t_j in the topic of a document d_i is a measure of the quality of t_j as a descriptor of documents similar to d_i. As we informally formulated earlier, a term is a good discriminator of a topic if it "tends to occur *only* in documents associated with that topic." We define

the *discriminating power of a term in the topic of a document* as a function $\Delta : \{t_0, \ldots, t_{n-1}\} \times \{d_0, \ldots, d_{m-1}\} \rightarrow [0, 1]$ calculated as follows:

$$\Delta(t_i, d_j) = \sum_{\substack{k=0 \\ k \neq j}}^{m-1} [[\delta(t_i, d_k)]^2 \cdot \sigma(d_k, d_j)].$$

Thus the discriminating power of term t_i in the topic of document d_j is an average of the similarity of d_j to other documents discriminated by t_i. The following are the topic descriptive and discriminating power for the terms in our example:

	$\Lambda(d_0, t_j)^T$	$\Delta(t_i, d_0)$
java	0.385	0.493
machine	0.158	0.524
virtual	0.014	0.566
language	0.040	0.517
programming	0.055	0.566
coffee	0.089	0.385
island	0.064	0.385
province	0.040	0.385
jvm	0.032	0.848
jdk	0.124	0.848

Guided by the notions of topic descriptors and discriminators, it is possible to learn novel context-specific terms and reinforce the weights of existing ones. This results in a better representation of the user search context, facilitating query refinement and context-based filtering. The next section shows how the proposed approach is applied in the design of an agent system for context-based Web retrieval.

3 AGENT SYSTEM ARCHITECTURE FOR CONTEXT-BASED WEB RETRIEVAL

An incremental approach to identify context-specific terms allows to go beyond the known user desires, to automatically generate a richer context representation through the use of topic descriptors and discriminators, and find what might be useful for the user. This kind of incremental mechanism can reveal similarities that were not previously apparent and present a "big picture" that can give the user a broader understanding of the current task.

Direct manipulation search interfaces provide fast access to information available on the Web, taking advantage of information indexed by major search engines (e.g., Google) or other searchable databases (PubMed, Amazon, etc.). However, the direct manipulation approach has a number of limitations including a large search space, actions in response to immediate user interaction only, and inability to learn from repetitive actions. An information agent-oriented approach can overcome some of these limitations, providing search and filtering capabilities, proactivity, task orientation and adaptivity [7].

The major challenge that an information agent that operates on top of a search interface needs to address is the generation of suitable queries. Our methods focus on how to incrementally generate queries based on context. Because search engines restrict queries to a small number of terms (e.g., the 32-term limit for Google) a single query cannot reflect extensive contextual information. In an incremental method, the first query terms generated for guiding a Web search may not provide the definitive results. However, comparing the set of search results to the user context can help to automatically refine subsequent queries.

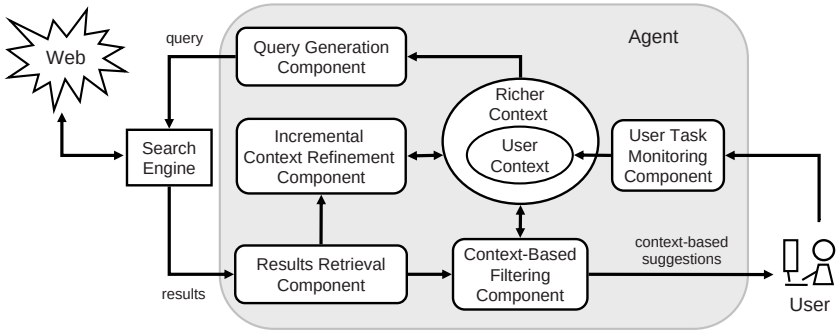


Figure 1: Agent system architecture for incremental context-based Web retrieval

Figure 1 depicts our agent system architecture for incremental context-based Web retrieval. In our prototype system, the agent handles partially observable environments such as the Web (through a search interface) and the user information needs (by observing the user behavior). It will maintain an internal state containing a representation of the user context, which is incrementally refined. The components that enable the agent to generate context-based suggestions are the following:

- **User Task Monitoring Component**. This component is in charge of observing how the user interacts with different kinds of computer utilities such as email systems, browsers and text editors, and generates a representation of the user context.
- **Query Generation Component**. This component selects terms from the user context and forms suitable queries, which are submitted to a standard search engine (e.g, Google) or entered into individual search forms (e.g., Amazon or PubMed). Initial queries will be entered by the user, or automatically formed with terms that occur frequently in the user context. Subsequent queries are refined as topic descriptors and discriminators are identified by the incremental search method.
- **Results Retrieval Component**. This component is in charge of retrieving the search results generated by the search interface, so that they can be locally analyzed.
- **Context-Based Filtering Component**. This component is in charge of estimating the relevancy of the documents collected by the results retrieval component. This is done by computing the similarity between the collected material and the user context. Both relevant and irrelevant material will be kept for use by the incremental context-refinement module (described below). However, only the material that is relevant to the current context will be presented as suggestions to the user.
- **Incremental Context-Refinement Component**. This component uses the content returned by the results retrieval component in combination with the relevancy information provided by the context-based filtering module to incrementally refine the context representation. It does so by adjusting the weights of the context-specific terms according to their descriptive and discriminating power.

4 EVALUATION

Because the goal of an information agent is to present useful suggestions, the ideal method for evaluating result quality would be an end-to-end user study, in which subjects directly assess the usefulness

of agent suggestions. However, to guide the bottom-up development of the methods, it is crucial to be able to assess incremental steps for which human-subjects evaluations would be impractical.

In previous work we showed that topic descriptors can help achieve good recall, while topic discriminators improve precision [18, 17]. Here, we present new empirical evidence that points out to the usefulness of combining topic descriptors and discriminators so that an intelligent agent performing context-based Web Search can maximize the contribution of both to form suitable queries.

In our evaluations we selected 15 pages from the ODP[1] directory (http://dmoz.org) to use as a base text representing the user context. These pages were obtained from the topics *Recreation*, *Society* and *Business* (we used 5 pages for each of the 3 topics). For each of the selected pages we generated a TF vector representation C. Then we applied an Intelligent Incremental Method (IIM) to generate queries. A schematic illustration of the IIM method is shown in figure 2 and summarized in the following steps:

1. Generate *k* queries using *l* terms in C, Queries[1..k], randomly;
2. i $\leftarrow$ 0;
3. For each *Query* in Queries, send it to a search engine;
4. Obtain the answers and convert the results to a vector representation;
5. Generate a sorted list L_Λ of topic descriptors;
6. Generate a sorted list L_Δ of topic discriminators;
7. Update L_Λ and L_Δ incrementally with some learning rate α;
8. Generate *k* queries using some combination of *m* terms from L_Λ and *n* terms from L_Δ, using a roulette selection method;
9. For each *Query* in Queries, send it to the search engine;
10. Obtain the answers and convert the results to a vector representation;
11. For each query result, test if it is a "good query";
12. For each bad query, reformulate it, obtain their results and convert them to a vector representation;
13. i $\leftarrow$ i + 1;
14. go to 5

In the IIM method, a good query is one that complies with at least one of the following conditions, depending on the case under analysis:

1. It retrieves at least one result whose similarity with C is higher than the highest similarity obtained in the previous iteration.
2. The average similarity of its' results is higher than the average similarity of the results from the previous iteration.

We have used Google Web API to collect search results and only the "snippets" returned by Google are used by our methods. The snippet is a text excerpt from the page summarizing the context in which the search terms occur. The input parameter l in the above algorithm determines the initial query size. The parameters m and n specify the number of descriptors and discriminators, respectively, used to form each of the subsequent queries. Note that this method generalizes those in which only descriptors or only discriminators are used (since either m or n could be set to zero). In our tests we tried different settings for the parameters l, m and n. The results reported here are restricted to the cases in which $l = 6$ and $m + n = 6$.

[1] Open Directory Project.

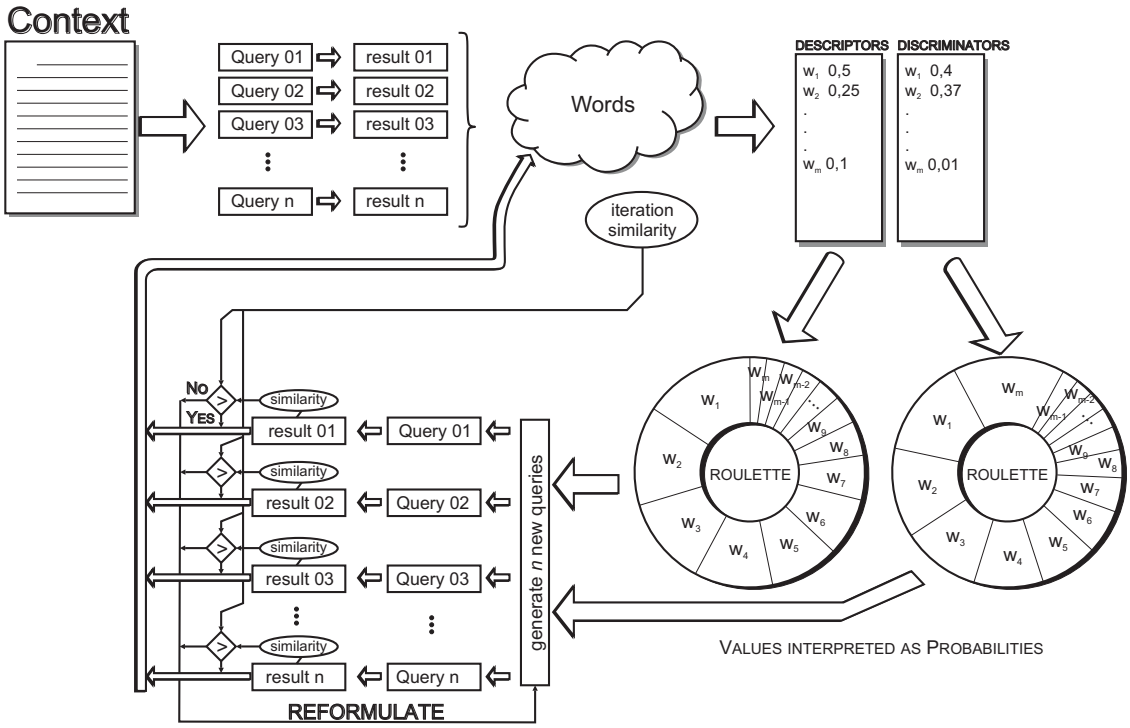


Figure 2: A schematic illustration of the IIM method.

As a baseline for comparison we used a Naïve Method (NM) that extends the baseline mechanism presented in [16] by supporting multiple simultaneous queries. In the same way as the IIM method, the NM method generates the initial queries randomly, and for the next iterations it generates the queries using a roulette-wheel method. We decided to use the roulette-wheel method due to its' simplicity, allowing us to favor the selection of the most promising terms. To set the probability of each term we used the term's weight as a descriptor or as a discriminator, depending on the case.

In our evaluation we take the minimum, average, and maximum similarity between the context (base document) and the retrieved results (snippets) as indicators of the methods' performance. The similarity measure is computed as the traditional cosine similarity in term space [2], except that the terms that occur in the query are disregarded. This prevents biasing the results in favor of those that select query terms directly from the base document.

Figure 3 shows the performance of the IIM and NM methods based on the minimum and average similarity between the search results and the initial context as an estimation of query quality (for the maximum similarity, the highest values achieved were approximately 0.6, starting from the initial iterations, and are not shown in the figure).

In our evaluations we observe that (1) NM always found the best results during the few first iterations, (2) IIM shows statistically significant improvement over the NM method, (3) IIM shows statistically significant improvements when the last iterations are compared to the first one, (4) IIM reduces its variance values at each iteration.

We performed additional evaluations, using different settings for the number of descriptors and discriminators in each query. Table 1 shows the performance based on average similarity for the three tested combinations of descriptors and discriminators (values in bold denote the maximum in that iteration):

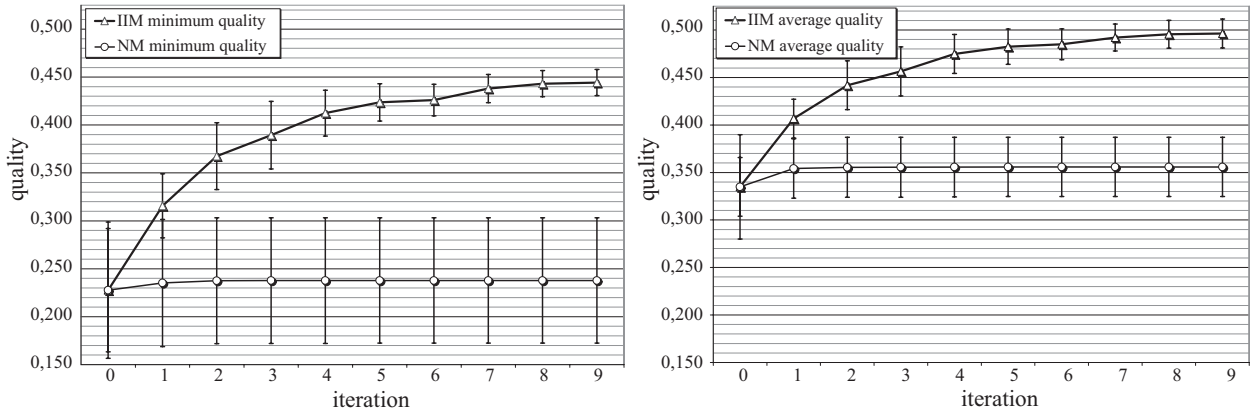


Figure 3: Performance Evaluation based on minimum similarity (left) and average similarity (right)

$\Lambda - \Delta$	(2–4)	(4–2)	(5–1)
iteration 0	0,328	**0,367**	0,335
iteration 1	0,351	**0,432**	0,407
iteration 2	0,367	0,441	**0,442**
iteration 3	0,374	0,449	**0,456**
iteration 4	0,388	0,454	**0,475**
iteration 5	0,400	0,467	**0,482**
iteration 6	0,408	0,479	**0,485**
iteration 7	0,413	0,482	**0,492**
iteration 8	0,415	0,488	**0,495**
iteration 9	0,419	0,495	**0,496**

Table 1: An analysis of the quality obtained using different combinations of descriptors and discriminators.

As stated in step 7 of our algorithm, the descriptor and discriminator weights were incrementally updated at each iteration based on some learning rate α as follows:

$$TermWeight(t,i) = TermWeight(t,i-1) * (1-\alpha) + NewTermWeight(t) * \alpha \quad (1)$$

We performed a number of tests to adjust the parameter α (not shown here for space reasons). For the cases analyzed, we found out that $\alpha = 0.4$ results in the best performance.

Another parameter tested is the number of queries generated during each iteration. We decided to use 20 queries for each generation but other settings could be tested as well.

5 RELATED WORK

Several existing systems exploit user interaction with computer applications to determine the user's current task and contextualize information needs. WebWatcher [1] is an early attempt to assist users locating information on the Web by highlighting hyperlinks in a page based on the declared preferences and browsing history of a user as well as information gathered from other users with similar interests. Personal WebWatcher [19] is a successor of WebWatcher that learns individual users' interests by observing their browsing behavior. Letizia [15] is a user interface agent that unobtrusively assists Web browsing. As the user navigates Web pages, Letizia performs a breadth-first search augmented by several heuristics to anticipate what items may be of interest to the user. Syskill & Webert

[20] uses information retrieval techniques to process the content of a page rated by a user, and machine learning to acquire a model, that is utilized to predict which links on a Web page a user will find useful. SenseMaker [4] is an interface that facilitates the navigation of information spaces by providing task specific support for consulting heterogeneous search services. The system helps users to examine their present context, move to new contexts or return to previous ones. Fab [3] is a hybrid content-based, collaborative Web page recommender system that learns to browse the Web on behalf of a user. Fab generates recommendations by the use of a set of collection agents (that find pages for a particular topic) and selection agents (that find pages for a particular user). Users' explicit ratings of the recommended pages combined with several heuristics are used to update personal-agents' profiles, remove unsuccessful agents, and duplicate successful ones. Broadway [12] is a case-based reasoning system that monitors a user's browsing activity and provides advice by reusing navigational cases extracted from past browsing experiences of a group of users. Another Web navigation assistant is SiteSeer [22], which recommends pages collaboratively by looking at users' bookmarks. The Watson system [8] is a context-aware search tool that attempts to find relevant online resources. Watson is part of a family of programs known as *Information Management Assistants* (IMAs) developed at the InfoLab of Northwestern University (Chicago, USA). The purpose of the IMAs is to anticipate the user's needs and to provide proactive and on demand support for the user's current activity. All these systems are similar to our proposal in attempting to provide users with context-relevant information, but differ in not attempting to learn context-specific terms by performing incremental search.

Extensions to basic IR approaches have examined some of the issues raised in this paper. For instance, some automatic relevance feedback techniques, such as the Rocchio's method [21], make use of the full search context for query refinement. In these approaches the original query is expanded by adding a weighted sum of terms corresponding to relevant documents, and subtracting a weighted sum of terms from irrelevant documents. As a consequence the terms that occur often in documents similar to the input topic will be assigned the highest rank, as in our descriptors. However, our technique also gives priority to terms that *occur only in relevant documents* and not just to those that *occur often*. In other words, we prioritize terms for both discriminating and descriptive power. The techniques for query term selection proposed in this paper share insights and motivations with other methods for query expansion and refinement [24, 6]. However, systems applying these methods differ from our framework in that they support this process through a query or browsing interface requiring explicit user intervention, rather than formulating queries automatically.

Our techniques rely on the notions of document similarity to discover higher-order relationships in collections of documents. This relates to the use of LSA [11] to uncover the latent relationships between words in a collection. However, LSA's goal is to compute a matrix representing semantic distance between terms and documents, without identifying topic descriptors and discriminators.

6 CONCLUSIONS

In this paper we have presented a novel approach for learning context-specific terms on the Web. Based on this approach, an intelligent agent can take advantage of the information available in the user context to perform incremental Web search. We have shown that the user context can be usefully exploited to access relevant material. However, those terms that occur more frequently in the user context are not necessarily the most useful ones. In light of this we proposed an incremental method for context refinement based on the analysis of search results. We have also proposed to distinguish two natural notions, namely topic descriptors and topic discriminators.

Our evaluations show the effectiveness of incremental methods for query generation and refinement. We are currently working on integrating the proposed method with qualitative approaches such

as the ones discussed in [9, 10] for ranking results based on user preferences. We are also analyzing different strategies for helping the system keep its focus on the initial context after several incremental steps have taken place. As part of our future work we expect to use standard large-scale collections (such as the TREC Web collection) to further evaluate our techniques.

Information agents have become essential components of today's Internet infrastructure. In light of the difficulties in accessing Web resources through traditional IR techniques, it is important to propose methods that allow the dynamic identification of useful context-specific material. We hope the methods proposed in this work provide new insights for further studies into this area.

REFERENCES

[1] Robert Armstrong, Dayne Freitag, Thorsten Joachims, and Tom Mitchell. WebWatcher: A learning apprentice for the World Wide Web. In *AAAI Spring Symposium on Information Gathering*, pages 6–12, 1995.

[2] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[3] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[4] Michelle Q. Wang Baldonado and Terry Winograd. SenseMaker: an information-exploration interface supporting the contextual evolution of a user's interests. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 11–18. ACM Press, 1997.

[5] Nicolas J. Belkin. Helping people find what they don't know. *Commun. ACM*, 43(8):58–61, 2000.

[6] Bodo Billerbeck, Falk Scholer, Hugh E. Williams, and Justin Zobel. Query expansion using associated queries. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 2–9. ACM Press, 2003.

[7] Jeffrey M. Bradshaw. An introduction to software agents. In Jeffrey M. Bradshaw, editor, *Software Agents*, pages 3–46. AAAI Press / The MIT Press, 1997.

[8] Jay Budzik, Kristian J. Hammond, and Larry Birnbaum. Information access in context. *Knowledge based systems*, 14(1–2):37–53, 2001.

[9] Carlos Chesñevar, Carlos Lorenzetti, Ana Maguitman, Fernando Sagui, and Guillermo Simari. Exploiting user context and preferences for intelligent web search. In *Proceedings of the Workshop de Investigadores en Ciencias de la Computación (WICC 2006)*, May 2006.

[10] Carlos Chesñevar and Ana Maguitman. ArgueNet: An argument-based recommender system for solving Web search queries. In *Proceedings of the International IEEE Conference on Intelligent Systems (IS 2004)*, pages 282–287. IEEE, June 2004.

[11] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[12] Michel Jaczynski and Brigitte Trousse. BROADWAY: A World Wide Web browsing advisor reusing past navigations from a group of users. In *Proceedings of the Third UK Case-Based Reasoning Workshop (UKCBR3)*, Manchester, UK, September 1997.

[13] Mei Kobayashi and Koichi Takeda. Information retrieval on the Web. *ACM Comput. Surv.*, 32(2):144–173, 2000.

[14] David B. Leake, Travis Bauer, Ana Maguitman, and David C. Wilson. Capture, storage and reuse of lessons about information resources: Supporting task-based information search. In *Proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems. Austin, Texas*, pages 33–37. AAAI Press, 2000.

[15] Henry Lieberman. Letizia: An agent that assists Web browsing. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. IJCAI-95*, pages 924–929, Montreal, Quebec, Canada, 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[16] Carlos Lorenzetti, Fernando Sagui, Ana Maguitman, Carlos Chesñevar and Guillermo Simari. Incremental Methods for Context-Based Web Retrieval. In *Proceedings of the Congreso Argentino de Ciencias de la Computación (CACIC 2006)*, October 2006.

[17] Ana Maguitman, David Leake, and Thomas Reichherzer. Suggesting novel but related topics: towards context-based support for knowledge model extension. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 207–214, New York, NY, USA, 2005. ACM Press.

[18] Ana Maguitman, David Leake, Thomas Reichherzer, and Filippo Menczer. Dynamic extraction of topic descriptors and discriminators: Towards automatic context-based topic search. In *Proceedings of the Thirteenth Conference on Information and Knowledge Management (CIKM)*, Washington, DC, November 2004. ACM Press.

[19] D. Mladenic. Personal WebWatcher: Design and implementation. Technical report ijs-dp-7472, School of Computer Science, Carnegie-Mellon University, Pittsburgh, USA, 1996.

[20] Michael J. Pazzani, Jack Muramatsu, and Daniel Billsus. Syskill & Webert: Identifying interesting Web sites. In *AAAI/IAAI, Vol. 1*, pages 54–61, 1996.

[21] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The Smart retrieval system - experiments in automatic document processing*, pages 313–323. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[22] James Rucker and Marcos J. Polanco. Siteseer: personalized navigation for the Web. *Communications of the ACM*, 40(3):73–76, 1997.

[23] G. Salton and C. Yang. On the specification of term values in automatic indexing. *Journal of Documentation*, 29:351–372, 1973.

[24] Falk Scholer and Hugh E. Williams. Query association for effective retrieval. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 324–331. ACM Press, 2002.

Algoritmos Genéticos Guiados para Predicción de Incendios Forestales

Mónica Denham, Ana Cortés, Tomás Margalef, Emilio Luque
Departament d'Arquitectura de Computadors i Sistemes Operatius, E.T.S.E,
Universitat Autònoma de Barcelona, 08193-Bellaterra (Barcelona) España,
monica.denham@aomail.uab.es
{ana.cortes,tomas.margalef, emilio.luque}@uab.es

Abstract

In this work a methology to improve forest fire prediction is proposed. For this purpose, we used an already developed fire simulator, called *fireLib*. Since one of the reasons why fire simulators are not able to exactly predict the fire evolution is the simulator input uncertainties, we proposed a tuning process based on a genetic algorithm (GA) to obtain an enhanced set of input parameters that provides an improved simulation. In order to reduce the big search space over which the GA operates, we introduce the use of some knowledge to guide the GA operations. We proposed two methods: a computational scheme and an analytical one. Both methods have been compared againts not using any kind of knowledge and the results are very encouraging.

Keywords: Parallel computing, Forest fire prediction, Genetic algorithms, High performance computing.

Resumen

En este trabajo se propone una metodología para mejorar las predicciones de incendios forestales. Utilizando un simulador ya desarrollado, se utiliza un algoritmo genético para realizar una búsqueda efectiva de parámetros de entrada tal que estos parámetros logren una buena predicción. A esto se le agrega el uso de conocimiento disponible para guiar las operaciones del algoritmo genético y reducir el gran espacio de búsqueda donde opera dicho algoritmo. Se proponen 2 métodos para guiar dicho algoritmo, el método computacional y el método analítico, este último además intenta verificar los resultados del método computacional. Dichos métodos se comparan mediante un estudio experimental mostrando los resultados la ganancia que supone añadir guía a la búsqueda frente a no añadirla.

Palabras claves: Procesamiento paralelo, Predicción de incendios forestales, Algoritmos genéticos, Cómputo de alto rendimiento.

1 INTRODUCCIÓN

En los últimos años se han perdido miles de hectáreas forestadas debido a incendios forestales. Estos incendios son una gran amenaza tanto a nivel ecológico como así también económico en todo el mundo.

Estos incendios tienen grandes impactos en la naturaleza: aumento de la desforestación del planeta, pérdida de zonas naturales que contribuyen a la calidad del aire, pérdida de ejemplares de árboles, vegetales y animales que habitan en los mismos, contaminación del aire por la generación de humo en la combustión, contaminación de cursos de agua cercanos a los mismos, cambios en las propiedades de la tierra, inundaciones (por pérdida de permeabilidad de la tierra), pérdida de vidas humanas, etc. Todo esto amenaza y diminuye la calidad de vida humana, animal y vegetal.

A nivel económico, se pierden recursos maderables lo cual influye en distintos tipos de industrias, se pierden tierras cultivables, etc.

Las causas de estos incendios son diversas, pueden ser incendios naturales (debido a conjunción de factores como temporadas muy secas, temporadas calurosas, etc.) pero la mayoría son producidas por factores humanos (un 90% aproximadamente [5]).

En la actualidad existe gran esfuerzo por parte del hombre en la lucha contra estos incendios. Esta lucha se realiza desde dos aspectos: prevención y lucha contra el fuego en un incendio.

La tecnología es una herramienta muy importante tanto para la prevención como para la lucha contra el fuego en el momento de un incendio. Existe mucho trabajo en este campo, obteniéndose herramientas cada vez más precisas y eficaces. La informática no es la excepción a esto y hoy en día existen diversos simuladores de comportamiento del fuego. Estos simuladores suelen tomar como entradas el estado del frente del fuego en un instante de tiempo inicial, y el estado del entorno donde se desarrolla el incendio y tienen como salida el estado del incendio en un instante de tiempo posterior al inicial. Estos simuladores pueden variar en el formato de sus entradas y sus salidas, el tipo de usuario al que está dirigido, el o los modelos que implementan, etc. Pero el problema de la mayoría de estos simuladores es que no siempre sus resultados se corresponden con la real propagación del fuego.

Una de las fuentes de error son sus parámetros de entrada. Estos parámetros describen el tipo de combustible, el clima y la topografía del lugar donde se desarrolla el fuego. Algunos de estos parámetros no varían o lo hacen muy poco en tiempo y espacio (pendiente, el tipo de vegetación, por ejemplo). En cambio el viento, la humedad del combustible, son más dinámicos y pueden cambiar con el ciclo día-noche y con el clima. Esta última característica hace que sea mas difícil disponer del valor correcto para estos parámetros en el momento de un incendio [5]. Además hay parámtros que no se pueden medir por lo tanto se deben utilizar estimaciones de los mismos, otros que deben ser interpolados ya que no se pueden medir en todos los puntos necesarios, los datos del terreno seguramente se actualicen con una cierta peridiocidad por lo que es normal trabajar con datos antiguos, etc. [1].

En este trabajo, se intentará mejorar la precisión de las predicciones intentando mejorar la precisión en los valores de los parámetros de entrada: se intentará encontrar el mejor valor para cada uno de los parámetros para que, utilizados en el simulador, la simulación resultante sea correcta.

Es importante notar que los parámetros son numerosos y que cada uno de ellos tiene un dominio propio por el que puede variar. Por lo tanto, encontrar el mejor valor para cada uno de ellos se vuelve un trabajo muy complejo dado que la cantidad de combinaciones posibles es muy alto. A esto hay que agregar que, al tratarse de una predicción, la misma está limitada en el tiempo para que realmente sea útil. Entonces, es necesario hacer la búsqueda de forma eficiente y rápida para que la misma no penalice el tiempo del proceso de predicción.

Este trabajo se enmarca dentro de la ciencia computacional ya que utiliza las características físicas

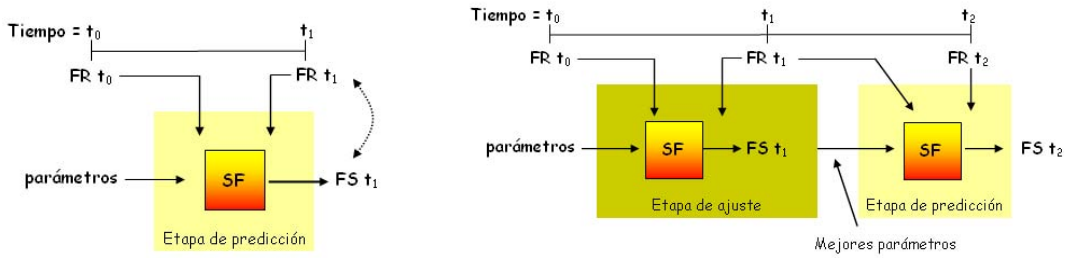


Figure 1: (a) Método clásico de predicción. (b) Método de predicción guiada por los datos.

del fuego que definen su forma de propagación, un modelo matemático que representa este comportamiento y distintas herramientas informáticas para implementar dicho modelo y utilizarlo en simulación de incendios forestales. Además, se utilizarán otras herramientas informáticas para optimizar la manipulación de información, y se utilizará dicha información para guiar la búsqueda del algoritmo genético y converger de forma rápida a buenas soluciones.

En la sección siguiente se mostrará la predicción clásica y la predicción en dos etapas la cual mejora a la predicción clásica. En la sección 3 se mostrará la forma en que se obtiene conocimiento a partir del mapa disponible del progreso del incendio hasta un instante de tiempo determinado. En la sección 4 se comentarán las principales características del algoritmo genético y los dos métodos propuestos para guiar la búsqueda de dicho algoritmo. La sección 5 mostrará la experimentación y los resultados obtenidos, y por último, la sección 6 muestra las conclusiones.

2 PREDICCIÓN DE INCENDIOS FORESTALES

Clásicamente, la predicción de incendios se realiza utilizando algún simulador de fuego (SF) [3] [6] [8] [7] [4] y como entradas se utilizan el estado del incendio real (FR) en el instante inicial (lo llamamos t_0) y un conjunto de parámetros y como resultado se obtiene el estado del frente del fuego en un instante de tiempo posterior (FSt_1). Este método clásico de simulación se muestra en la figura 1 (a).

Una vez que se obtiene la predicción (FSt_1) se compara con el estado del frente del fuego real (FRt_1) para medir la calidad de la predicción. Este método tiene como ventajas la simplicidad y que los recursos que necesita son pocos (tanto en información como en procesamiento). Pero tiene la gran desventaja de depender de la calidad de la única combinación de parámetros de entrada que se utiliza. Y como hemos mencionado anteriormente, no siempre se dispone de los valores correctos para los parámetros.

En este trabajo se utiliza una estrategia de predicción que pretende mejorar el método clásico. En este método, se agrega una etapa previa a la etapa de predicción para encontrar distintas combinaciones de parámetros de entrada que logren buenas simulaciones. En esta etapa, la calidad de la simulación obtenida para una determinada combinación de los parámetros de entrada del simulador determinará una retroalimentación que permitirá mejorar paso a paso los valores de los parámetros.

Este método está formado por 2 etapas (figura 1 (b)): etapa de ajuste y de predicción. En la etapa de ajuste, se trata de determinar valores para los parámetros tales que logren una buena simulación del instante de tiempo t_0 a t_1. Para esto, se utiliza el simulador con la línea de fuego en el instante t_0 (FRt_0) y los parámetros de entrada. El simulador retorna la línea de fuego simulada para el instante t_1 (FSt_1). Este línea de fuego es comparada con la línea de fuego real en el instante t_1 (FRt_1) y, dependiendo del resultado de esta comparación, se irán buscando los valores para los parámetros que

mejoren los resultados de la simulación. Una vez concluída la etapa de ajuste, se elije el conjunto de parámetros que logren la mejor simulación para t_1 y se utilizará como entrada del simulador junto con la línea de fuego real en el instante t_1 (FRt_1) para predecir el estado del incendio en el próximo instante de tiempo t_2 (FSt_2).

Esta forma de enfocar la predicción en 2 etapas, fue inicialmente definida por B. Abdalhaq en [1]. Sin embargo, en dicho trabajo la etapa de ajuste no estaba diseñada para permitir guiar la sintonización de los parámetros de entrada en función del conocimiento de la evolución del fuego real. Nuetra propuesta supone mejorar la etapa de ajuste acotando el espacio de búsqueda en función del conocimiento extraído de la propia evolución del fuego real. La estrategia escogida para obtener dicho conocimiento se describe en la siguiente sección.

3 OBTENCIÓN DEL CONOCIMIENTO

El modelo matemático de comportamiento del fuego propuesto por Rothermel [9] es uno de los modelos más utilizados en este campo. La mayoría de los simuladores de incendios forestales implementan dicho modelo [5]. Éste calcula el índice de máxima propagación y la intensidad de reacción del fuego a partir de las características del entorno donde se desarrolla el fuego.

Dicho modelo tiene como parámetros las características del entorno donde se desarrolla el fuego, las características del clima y de la vegetación. En este trabajo se utiliza el simulador *fireLib* que implementa el modelo matemático de Rothermel y que tiene como parámetros: grados de inclinación y orientación de la pendiente, tipo de vegetación que actuará como combustible, humedad del combustible herbáceo vivo, humedad del combustible herbáceo muerto (en tres momentos distintos), dirección y velocidad en que corre el viento [4].

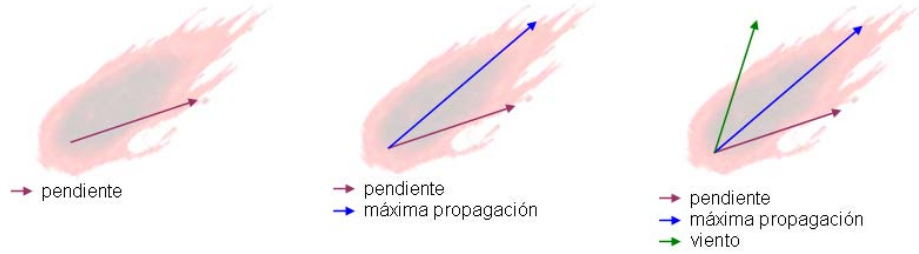
De los distintos parámetros del modelo matemático de comportamiento del fuego, el viento y la pendiente determinan la dirección de máxima propagación del fuego. La pendiente es un parámetro que se asume que no varía en espacio y tiempo y que es un dato que se conoce al momento de la simulación. En cambio el viento, es uno de los parámetros más dinámicos, cambia constantemente y en el microclima generado por un incendio lo hace de forma aún más significativa. La dirección y velocidad del viento son los parámetros de entrada de los cuales se intenta encontrar su valor más preciso para generar una buena simulación.

Como se puede observar en la figura 1 (b) una entrada al método es el mapa del incendio en el instante t_1 (FRt_1). Esta información es requerida por el método por lo tanto se dispone de ella. Entonces, se puede utilizar el conocimiento que brinda el avance del fuego hasta el instante t_1 con el objetivo de descubrir las características de la propagación real (dirección y velocidad de máxima propagación). Entonces, con estos datos, se puede descubrir las características del viento que, compuestas con las características de la pendiente, obtengan la propagación observada. En la figura 2 (a), (b) y (c) se muestran los pasos de dicho proceso.

En este trabajo se utilizará un algoritmo genético para intentar realizar la búsqueda de una forma efectiva. Además, se intentará guiar dicho algoritmo basándonos en el conocimiento aportado por el análisis del mapa hasta el instante t_1 como se explicó anteriormente. En la próxima sección se explicarán las principales características del algoritmo genético utilizado y cómo se realizará la guía para obtener mejores resultados y de forma más rápida.

4 ALGORITMO GENÉTICO GUIADO

El algoritmo genético se basa en la evolución natural, donde se tiene una población de individuos la cual evoluciona sobreviviendo los individuos mejor adaptados a su ambiente. Para obtener las



(a) (b) (c)

Figure 2: (a) Mapa real t_1 y efecto de la pendiente. (b) Obtención de las características de máxima propagación a partir del mapa en el instante t_1. (c) Cálculo del efecto del viento necesario para dicha propagación y la pendiente específica para el caso.

sucesivas generaciones se realizan tres operaciones sobre los individuos: *selección*, *cruzamiento* y *mutación*. Dichas operaciones hacen que las propiedades de los individuos bien adaptados al ambiente se transmitan de generación en generación, obteniendo así individuos los cuales tienen cada vez mejores propiedades. Existen diversas técnicas para cada una de estas operaciones. A continuación se explicará muy brevemente cómo se realiza cada una de ellas en este trabajo.

En nuestro dominio de trabajo, un individuo es una asignación específica de valores para cada uno de los siguientes parámetros: humedad del combustible vivo, humedad del combustible muerto pasada 1 hora, pasadas 10 horas y pasadas 100 horas, y dirección y velocidad del viento. Una población será un conjunto de individuos. Para nuestra aplicación, que un individuo esté bien adaptado a su ambiente es que el mapa resultante de la simulación utilizando dicho individuo de un mapa similar al mapa real. Dado que *fireLib* identifica los terrenos como mapas divididos en celdas, nuestra función de semejanza determina el grado de similitud del mapa real y el simulado en base a las celdas diferentes de ambos mapas. Dicha función la llamaremos función de error.

En la *selección* se elijen los individuos que se utilizarán como padres para crear a los individuos hijos que formarán parte de la próxima generación. Esta operación tiene como propiedad que los individuos mejor adaptados a su ambiente tienen más probabilidades de ser elegidos. A esta operación se le puede agregar *elitismo* donde los mejores j ($j > 0$) individuos forman parte directamente de la próxima generación.

La operación de *cruzamiento* se utiliza para obtener nuevos individuos a partir de los individuos padres. Se define un punto de cruzamiento aleatorio y se intercambian los valores de los padres a partir de dicho punto para formar a los 2 hijos.

La *mutación* consiste en modificar el valor de alguna de las características por algún otro valor válido. Esto se realiza bajo una probabilidad y para cada una de las características de los individuos hijos. Como en la naturaleza, esta probabilidad es baja.

El objetivo de estas 2 últimas operaciones es garantizar variabilidad y evitar caer en máximos o mínimos locales.

Como se propone en la sección anterior, se utilizará el conocimiento aportado por el avance del fuego real para determinar las características ideales del viento para lograr la propagación real. Se modificarán las operaciones de elitismo y de mutación para forzar la asignación de valores que ayudarán a encontrar la misma dirección y velocidad de propagación que la encontrada en el mapa real.

En este trabajo se proponen 2 técnicas para guiar la búsqueda: computacional y analítico. En las próximas subsecciones se explicarán brevemente cada uno de ellos.

4.1 Método Computacional

Este método utiliza información de distintas propagaciones para distintas configuraciones de viento, pendientes y humedades para generar una base de datos de casos genéricos de propagación. Con esta información se elije el comportamiento del fuego que más se aproxime a la propagación real observada hasta t_1 para una pendiente determinada y utiliza las características del viento para guiar el algoritmo genético.

Esta técnica es totalmente independiente del simulador utilizado. La obtención de las mejores caracteristicas del viento se realiza una sola vez al comienzo de la etapa de ajuste, antes de evolucionar la población.

Como en la base de datos no se puede tener todos los casos de propagaciones, se tiene en cuenta que los datos retornados corresponden a la propagación más parecida (difícilmente sea igual) por lo que los datos retornados no serán utilizados como valores absolutos, sino que definirán un subrango dentro del rango válido para la dirección y la velocidad del viento.

La figura 3 muestra algunos registros de la base de datos que utiliza el método. Los datos almacenados son: modelo de vegetación, pendiente del terreno, dirección de propagación del fuego, velocidad de propagación del fuego, distancia recorrida, y por último los datos del viento: dirección y velocidad.

Si por ejemplo se tiene un fuego en un terreno con 45º de inclinación, modelo de vegetación 7, y el mapa muestra que la máxima propagación se produjo a 45º y a una velocidad de 20 pies por minuto, entonces, según los datos de la figura 3 el registro elegido muestra un viento de 9 pies por minutos a una dirección de 45º. Luego estos datos se utilizarán para acotar el rango de variación de la dirección del viento y el de su velocidad.

Estos rangos acotados se utilizararán en la operación de mutación del algoritmo genético, en vez de mutar los valores por otro aleatorio se tomará un número también aleatorio pero en un rango mucho más pequeño (10º aproximadamente) con centro en el valor retornado. Lo mismo sucede con la velocidad, la cual variará en un nuevo rango de amplitud de 4 millas por hora con centro igual al encontrado en la tabla. La amplitud de los nuevos rangos es configurable antes de cada simulación.

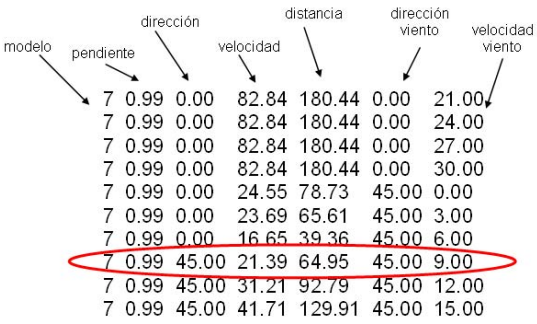
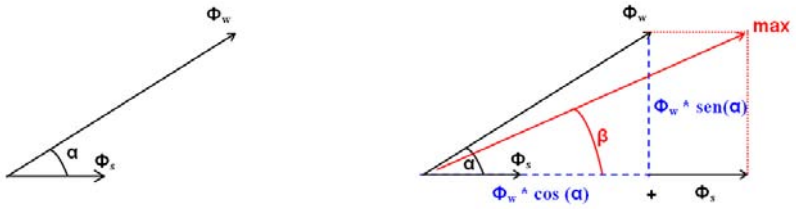


Figure 3: Datos almacenados en la base de datos para el método Computacional. La pendiente está expresada en radianes.

4.2 Método Analítico

El método analítico se basa en el conocimiento exhaustivo del modelo analítico de propagación del fuego implementado en el simulador para descubrir la dirección de máxima propagación del fuego con el objetivo de obtener los valores para el viento. Este método depende del simulador ya que se basa en sus operaciones y será más exacto ya que determinará para cada uno de los individuos la

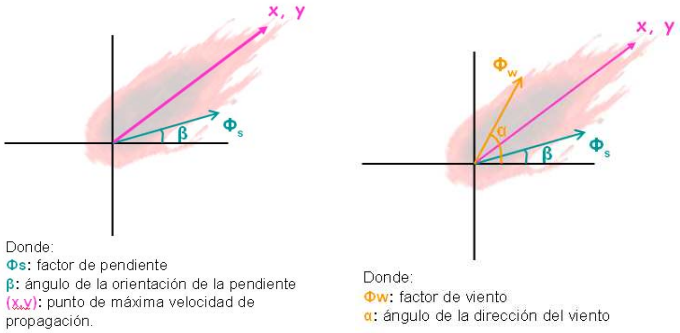


(a) b

Figure 4: (a) Vectores de efecto y direccción del viento y de la pendiente. (b) Composición del vector de propagación a partir de los vectores del viento y de la pendiente.

dirección y velocidad del viento ideales para lograr la propagación real. Este método se utiliza para validar al método computacional.

El estudio del simulador nos permitió observar cómo implementa el modelo de Rothermel y específicamente cómo determina la dirección y velocidad de máxima propagación. Sea Φ_w el efecto del viento, Φ_s el efecto de la pendiente y α la diferencia entre la dirección del viento y la dirección hacia donde sube la pendiente (figura 4 (a)). Por propiedades trigonométricas se suman ambos vectores para componer un tercer vector que determina el ángulo de máxima propagación y el índice de propagación (en la figura 4 (b) β y max respectivamente). Este nuevo vector tiene altura igual a $\sin(\alpha) * \phi_w$ y base igual a $\cos(\alpha) * \phi_w + \phi_s$, y ángulo β definido por estos dos vectores.



(a) (b)

Figure 5: (a) Conocimiento disponible: datos pendiente y propagación real. (b) Obtención del ángulo y efecto del viento.

La figura 5 muestra cómo se trabaja en este método. De la propagación real del fuego se obtienen los datos de la pendiente (β y Φ_s) y el punto (x, y) que determina el punto de máxima propagación. Del esquema de la figura 5 (b) se tienen las siguientes ecuaciones:

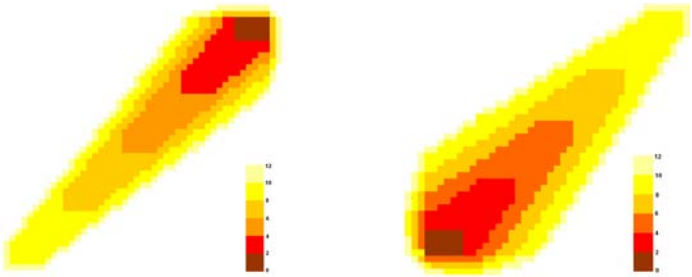
$$x = \Phi_s * \cos(\beta) + \Phi_w \cos(\alpha) \tag{1}$$

$$y = \Phi_s * \sin(\beta) + \Phi_w \sin(\alpha) \tag{2}$$

Entonces, como se conocen x e y, es posible despejar de las ecuaciones 1 y 2 los datos del viento:

$$\Phi_w = \frac{x - \Phi_w \cos(\beta)}{\cos(\alpha)} \tag{3}$$

$$\alpha = \arctan(\frac{y - \Phi_s \sin(\beta)}{x - \Phi_s \cos(\beta)}) \tag{4}$$



(a) (b)

Figure 6: Mapas utilizados para las pruebas. (a) Mapa sintético 1. (b) Mapa sintético 2.

Una vez que se obtiene el valor de Φ_w, este valor es utilizado para obtener la velocidad del viento (en millas por hora) que se debería utilizar como parámetro para llegar al efecto del viento Φ_w en la simulación.

Las figuras y las ecuaciones están esquematizadas y simplificadas. Es importante considerar que tanto el efecto del viento como de la pendiente están atenuados por factores que dependen del tipo de combustible, humedades del combustible muerto, humedad del combustible vivo, etc. El cálculo de las ecuaciones (3) y (4) está representado trigonométricamente, sin incluir los cálculos que hace el simulador al modelizar a través de vectores estos fenómenos (viento, pendiente y propagación del fuego). Además no está incluída la propagación sin considerar viento y pendiente que también influyen en la propagación final obtenida.

Entonces, cuando se obtiene la simulación con cada individuo, se determina qué características debería tener el viento en dicho individuo para llegar a la propagación real. Estas características luego se utilizarán para guiar la mutación o el elitismo en la próxima generación. Cabe notar que en el caso de la mutación guiada, los valores calculados para los individuos padres son utilizados en los hijos, lo cual puede ser una fuente de error. Esto no sucede con el elitismo guiado ya que los individuos no se cruzan.

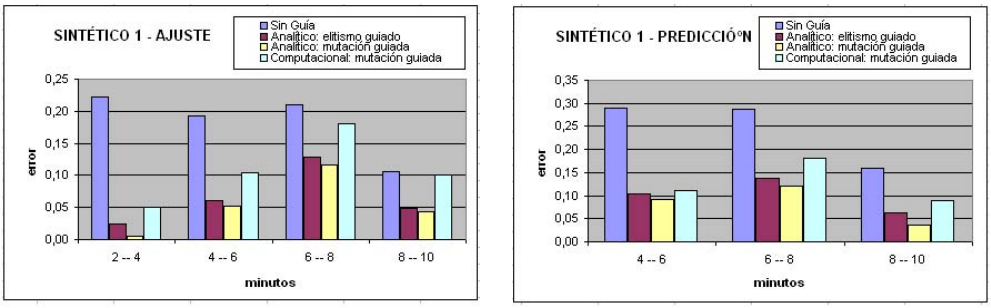
En la próxima sección se mostrarán resultados obtenidos con ambos métodos en dos casos de incendios.

5 EXPERIMENTACIÓN Y RESULTADOS

Las pruebas realizadas tienen como objetivo analizar la ganancia de agregar conocimiento para guiar la mutación en los dos métodos propuestos y, en particular, el elitismo en el método analítico, del algoritmo evolutivo.

Para la experimentación se han utilizado dos mapas sintéticos (resultados de simulaciones utilizando *fireLib*) con el fin de validar la metodología propuesta. Para cada mapa se consideran los pasos de simulación que dependen de la duración del incendio. En ambos casos, se evalúan los pasos del minuto 2 al 4, del 4 al 6, del 6 al 8 y del 8 al 10, para el paso de ajuste y del minuto 4 al 6, del 6 al 8, del 8 al 10 para el paso de predicción. En ambos mapas, se considera el tipo de vegetación donde predominan arbustos pequeños con muchas ramas (modelo 7 en [2]). Los mapas de evolución de estos fuegos sintéticos se muestran en las figuras 6 (a) y (b).

Ambas pruebas comparten algunas características: para garantizar un cierto grado de diversidad en los individuos de las poblaciones se comienzan las ejecuciones (paso de ajuste del minuto 2 al 4) con poblaciones de 50 individuos con valores aleatorios uniformemente distribuidos en sus rangos de variación. En los siguientes pasos de simulación se utiliza la población evolucionada en el paso de



(a) (b)

Figure 7: Mapa sintético 1: (a) Error paso de ajuste. (b) Error paso de predicción.

simulación anterior. Esto hace que se partan de individuos que han sido buenos en el paso anterior, los cuales deberían ser buenos en el paso siguiente de simulación. Además, los resultados obtenidos son el promedio de la ejecución del método para 5 poblaciones iniciales distintas.

Diversas pruebas del algoritmo genético permitieron observar que es en las primeras evoluciones donde el algoritmo se orienta a zonas del espacio de búsqueda con individuos que producen simulaciones con error bajo, y a partir de un cierto punto no conviene seguir evolucionando, pues el error decrece de forma insignificativa y no se justifica seguir iterando para obtener nuevas evoluciones. Por esto, se utilizan 5 pasos de evolución en cada una de las pruebas.

En el primer caso (mapa de la figura 6 (a)) el terreno es de 89 metros de ancho por 109 de largo y se encuentra en un terreno con pendiente de 18 grados. El terreno se divide en celdas de 1 metro cuadrado.

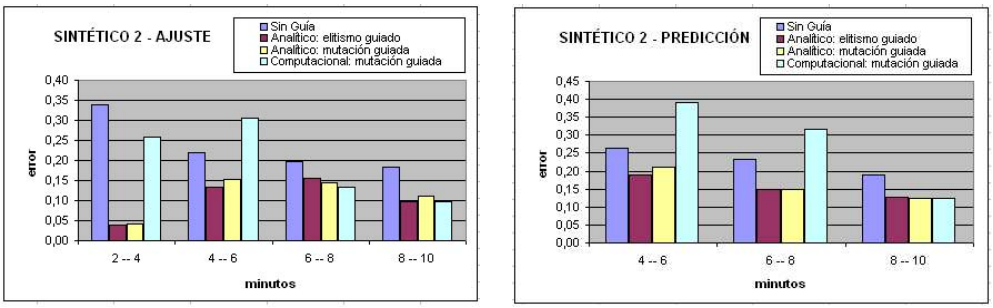
Las figuras 7 (a) y (b) muestran los resultados obtenidos con este mapa, en los pasos de ajuste y de predicción respectivamente. En ambos gráficos, el eje x muestra el paso de simulación y el eje y el error cometido con el mejor de los individuos de la población evolucionada. En los gráficos se muestran los resultados con los distintos métodos propuestos: algoritmo genético sin guiar la búsqueda, utilizando el método computacional para guiar la operación de mutación y utilizando el método analítico para guiar las operaciones de mutación y de elitismo.

Las figuras nos permiten ver que en todos los casos utilizar conocimiento para guiar la búsqueda disminuye el error tanto en los pasos de ajuste como en las predicciones. En todos los casos, el método analítico tiene mejores resultados si se los compara con el computacional, rasgo que es esperable ya que el método analítico calcula los mejores valores del viento para cada uno de los individuos de forma más precisa. En este caso se consideran todas las características del individuo que influyen en la propagación. En cambio en el computacional sólo se considera el avance del fuego real hasta el instante t_1.

En el segundo caso el fuego se ha desarrollado como muestra la figura (6 (b)). El terreno es de 33,22 metros de largo por 27,12 de ancho. En este caso, el mapa se divide en celdas más pequeñas, de 0,30 metros cuadrados (1 pié cuadrado). El terreno tiene 27 grados de pendiente.

Las figuras 8 (a) y (b) muestran los resultados de la misma forma a los resultados del primer mapa. En este caso, el método analítico mejora el error de los individuos en todos los casos. Pero el método computacional no logra buenos resultados en el paso 2 a 4 y 4 a 6 de la etapa de ajuste. Para entender este comportamiento no esperado, se analizarán los resultados obtenidos para cada una de las poblaciones en los pasos de ajuste y de predicción para dicho mapa.

Las figuras 9 (a) y (b) muestran los resultados de cada población del paso de ajuste y las figuras 10 (a) y (b) muestran los resultados para el paso de predicción. En estas figuras sólo se muestran los resultados para el método sin guiar y para el método computacional que es donde se han observado



(a) (b)

Figure 8: Mapa sintético 2: (a) Error paso de ajuste. (b) Error paso de predicción.

Sin Guía - Ajuste						
Paso	Pob 1	Pob 2	Pob 3	Pob 4	Pob 5	Promedio
2 - 4	0,49	0,57	0,29	0,20	0,15	0,34
4 - 6	0,39	0,26	0,15	0,19	0,11	0,22
6 - 8	0,44	0,22	0,07	0,18	0,07	0,20
8 - 10	0,45	0,16	0,05	0,16	0,09	0,18

(a)

Método Computacional - Ajuste						
Paso	Pob 1	Pob 2	Pob 3	Pob 4	Pob 5	Promedio
2 - 4	0,24	0,24	0,29	0,23	0,29	0,26
4 - 6	0,32	0,32	0,32	0,28	0,29	0,31
6 - 8	0,11	0,16	0,16	0,11	0,12	0,13
8 - 10	0,10	0,09	0,11	0,10	0,09	0,10

(b)

Figure 9: (a) Error para el paso de ajuste sin utilizar guía. (b) Error para el paso de ajuste utilizando el método computacional.

los comportamientos no esperados.

Entonces, observando los resultados para cada una de las poblaciones, se pudo observar que cuando no se guía la búsqueda, los resultados de las distintas poblaciones difieren mucho entre sí. Al contrario, cuando se guía la búsqueda con cualquiera de los métodos propuestos, el error cometido en las distintas poblaciones se mantiene estable entre las distintas poblaciones. Se puede observar cómo, si no se guía la búsqueda, se encuentran poblaciones con errores grandes y otras con errores muy pequeños, y estos últimos casos, disminuyen notablemente el error promedio obtenido.

Este análisis nos permite ver cómo cuando no se guía la búsqueda, la calidad de los resultados dependen de la suerte de la “bondad” tanto en los valores aleatorios de las poblaciones iniciales así como también de los valores aleatorios elegidos en las operaciones del algoritmo genético. Depender de esta suerte no es bueno en un problema como el nuestro donde la predicción quedaría sujeta a dicha suerte (¿qué ocurriría si, por ejemplo, trabajamos con una sóla población y dicha población tiene el comportamiento de la población 2 en el paso de ajuste?). Pues esto hace ver que en un problema de predicción como el nuestro, garantizar que el algoritmo no depende de valores aleatorios sino que la búsqueda se orienta hacia zonas con buenos individuos es una buena característica.

Las figuras 10 (a) y (b) muestran los resultados de cada población del paso de predicción. Estos casos tienen las mismas características que las explicadas para el paso de ajuste, ya que se utilizan los

Sin Guía - Predicción						
Paso	Pob 1	Pob 2	Pob 3	Pob 4	Pob 5	Promedio
4 - 6	0,39	0,39	0,24	0,20	0,10	0,26
6 - 8	0,43	0,25	0,17	0,25	0,07	0,23
8 - 10	0,44	0,16	0,09	0,17	0,09	0,19

(a)

Método Computacional - Predicción						
Paso	Pob 1	Pob 2	Pob 3	Pob 4	Pob 5	Promedio
4 - 6	0,38	0,38	0,38	0,38	0,44	0,39
6 - 8	0,34	0,37	0,37	0,28	0,23	0,32
8 - 10	0,10	0,12	0,21	0,09	0,10	0,12

(b)

Figure 10: (a) Error para el paso de predicción sin utilizar guía. (b) Error para el paso de predicción utilizando el método computacional.

mejores individuos de dicho paso para realizar la predicción, por lo que es normal que suceda esto.

Por otro lado, observando los resultados del paso de predicción, para los 2 primeros instantes se observa que al utilizar un individuo (malo) encontrado en la etapa de ajuste resulta una mala predicción. Por eso es normal los errores encontrados en las predicciones.

Analizando los resultados obtenidos, se puede determinar la ganancia (o pérdida) obtenida para cada uno de los métodos de guía con respecto al método sin guiar la búsqueda. Comparando los resultados de los 2 mapas, se determina que para el paso de ajuste, el método computacional tiene una reducción del 26,26% del error, el método analítico guiando la mutación tiene una reducción del error del 58,07% y guiando el elitismo, reducimos un 55,69% el error. A su vez, para el paso de predicción se tiene que se reduce un 15,24% usando el método computacional, un 48,91% guiando la mutación con el método analítico y un 45,50% guiando el elitismo también con este último método. Esto nos permite ver que ambos métodos reducen el error en los dos pasos del método de predicción (ajuste y predicción).

6 CONCLUSIONES

Este trabajo permitió ver que una de las causas de la falta de precisión de los resultados de los simuladores es la imprecisión de los parámetros de entrada. Por lo tanto, una de las formas de mejorar la calidad de las simulaciones consiste en mejorar la calidad de los parámetros de entrada.

En este trabajo se propone utilizar un algoritmo genético para realizar una búsqueda eficiente de buenos valores para los parámetros de entrada (valores que obtengan buenas simulaciones). A este algoritmo genético, se le agrega la utilización de conocimiento en el paso de ajuste para guiar la elección de valores en las operaciones del algoritmo genético. Estas modificaciones permiten guiar al algoritmo genético para que se exploren zonas del espacio de búsqueda que contengan buenos individuos y no explorar zonas que den simulaciones alejadas de la realidad.

Con estas guías se han hecho distintas experimentaciones para comparar los resultados. Estas experimentaciones han mostrado que el método computacional y el analítico dan buenos resultados, pues disminuyen en la mayoría de los casos el error de las simulaciones. El método analítico muestra mejores resultados lo cual es esperable dado que se calculan las características del viento para cada individuo. El procesamiento de esto no es más que realizar una secuencia de operaciones por

cada individuo después de cada simulación. A su vez, el método computacional agrega una fase de "searching" en una base de datos de comportamientos del fuego de forma eficiente para minimizar el impacto en tiempo de cómputo requeridos.

Los resultados obtenidos han mostrado que el comportamiento del método computacional sigue el mismo patrón de comportamiento que el método analítico lo que nos permite validar dicha estrategia.

Una característica que se pudo apreciar es que al no guiar la búsqueda, los resultados de las evoluciones utilizando distintas poblaciones difieren bastante en el error cometido. Esto no es favorable en un problema donde se buscan buenas predicciones, ya que al final la calidad de la predicción puede depender de la calidad de los valores elegidos aleatoriamente.

REFERENCES

[1] Abdalhaq B. A methodology to enhance the Prediction of Forest Fire Propagation. Tesis Doctoral. Universitat Autònoma de Barcelona (España). Junio 2004

[2] Hal E. Anderson. Aids to Determining Fuel Models For Estimating Fire Behaviour Intermountain Forest and Range Experiment Sation Ogden, UT 84401. General Technical Report INT.122. April 2002.

[3] Andrews P.L. BEHAVE: Fire Behavior prediction and modeling systems - Burn subsystem, part 1. General Technical Report INT-194. Odgen, UT, US Department of Agriculture, Forest Service, Intermountain Research Station. pp. 130. 1986.

[4] Bevins C. D. FireLib User Manual & Technical Reference. 1996. Disponible en: http://www.fire.org/downloads/fireLib/1.0.4/firelib.pdf. Accedido en enero de 2006.

[5] Bianchini G. Wildland Fire Prediction based on Statistical Analysis of Multiple Solutions. Tesis Doctoral. Universidad Autónoma de Barcelona. España. Julio de 2006.

[6] Finney M. A. FARSITE: Fire Area Simulator-model development and evaluation. Res. Pap. RMRS-RP-4, Ogden, UT: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. pp. 47. 1998.

[7] FIRE.ORG - Public Domain Software for the Wildland fire Community. http://www.fire.org. Accedido en mayo 2007.

[8] Reinhardt E.D., Keane R. E., Brown J.K. First Order Fire Effects Model: FOFEM 4.0, User's Guide. General Technical Report INT-GTR-344. 1997.

[9] Rothermel R. C. A mathematical model for predecting fire spread in wildland fuels. USDA FS, Ogden TU, Res. Pap. INT-115, 1972.

Alternativa de Comprobación sintáctica de VLP*: $Gsig\_Parsing$. Aspectos formales y el caso de estudio: $E\_GraPiCO$.

Carlos Andrés Tavera Romero
Universidad de San Buenaventura - Cali; Plan de Ingeniería de Sistemas
catavera@usb.edu.co

and

Juan Francisco Díaz Frías
Universidad del Valle; Escuela de Ingeniería de Sistemas
jdiaz@eisc.univalle.edu.co

and

Aybert Soto Vargas
Universidad Cooperativa de Colombia-Cali; Programa de Ingeniería de Sistemas
saybert01@yahoo.es

and

Javier Andrés Gallego Varona
Universidad Cooperativa de Colombia-Cali; Programa de Ingeniería de Sistemas
Javan_jl@yahoo.es

and

Ánderson Jojoa Giraldo
Universidad Cooperativa de Colombia-Cali; Programa de Ingeniería de Sistemas
anderson_jojoa@yahoo.es

Abstract

This publication presents $Gsig\_Parsing$, an alternative syntactic testing mechanism for visual languages through the division of parsing in two stages, context-free and context-dependent analisis. As a real example of the new proposed form, the visual programm editor E_GraPiCO is shown.

Keywords: Visual languages, Computational calculus, Parsing.

Resumen

En esta publicación se presenta $Gsig\_Parsing$, una alternativa de un mecanismo de comprobación sintáctica para lenguajes visuales por medio de la división del parsing en dos etapas, los Análisis Independientes de Contexto y los Sensibles al Contexto. Como un ejemplo real de la nueva forma propuesta se muestra el editor de programas visuales E_GraPiCO.

Palabras clave: Lenguajes Visuales, Cálculo computacional, Análisis sintáctico.

*Lenguajes de Programación Visual

1. Introducción

Cada vez que se requiere la implementación de un procesador de algún lenguaje, siempre se efectúan de alguna manera las etapas de lo que se entiende como compilador: Análisis léxico, Análisis sintáctico, Análisis semántico, Optimización y Generación de código.

Si se tienen n compiladores cruzados[1], y a su vez, escalonados[2] se tendrá una cantidad n de análisis léxicos, sintácticos, ... y, así sucesivamente, como se muestra en la figura 1.

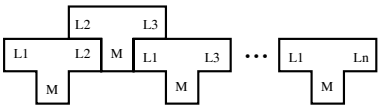


Figura 1: Compilador escalonado.

Aunque las etapas de análisis léxico y sintáctico son bien conocidas y su implementación no presenta un alto nivel de complejidad (por lo menos en las gramáticas de los lenguajes que están dentro de la clase 3 de la jerarquía de Noam Chomsky) prescindir de alguno de estos procesos puede representar una gran ayuda al momento de diseñar y/o implementar un procesador de lenguajes. Para dejar de efectuar el análisis léxico y/o el sintáctico en un compilador B que recibe código de otro A, se requiere la total certeza de que el código resultante de A esté léxica y sintácticamente correcto, además de que se debe contar en B con un mecanismo que aproveche esta característica.

Uno de los casos donde aporta mayor utilidad el renunciar a etapas de compilación, ya sea el análisis léxico o sintáctico, es en los Lenguajes Visuales, dado que para el almacenamiento de un programa se emplea un código intermedio con formato textual para poder realizar los análisis y la traducción a código objeto. Consecuente con lo anterior, es que si se efectua un análisis sintáctico analizando los íconos desde una gramática para lenguajes visuales (tarea de sobra costosa) posteriormente al realizar el proceso de compilación desde el programa almacenado textualmente, se deberá efectuar de nuevo el análisis sintáctico y después el resto de los análisis y procedimientos. Uno de los mecanismos existentes más difundidos para realizar el análisis sintáctico de los Lenguajes Visuales es el presentado en [2], consiste en un mecanismo para verificar la sintaxis de un programa visual mediante una forma especial del conocido análisis LR(0), con los inconvenientes que éste pueda tener, como el requerimiento de la construcción de una tabla de análisis sintáctico (que resulta ser bastante grande como se puede apreciar en el artículo). De otro lado, si no se tome en la cuenta las características operativas del análisis sintáctico de un programa visual, resta la dificultad de extraer cada uno de los sintagmas[3] presentes.

Surgen entonces, dos interrogantes:
¿Cómo efectuar el análisis sintáctico en el código visual sin tanto costo?
¿Cómo implementar procesos de compilación como la traducción desde un programa visual hacia algún tipo de código textual sin repetir el análisis sintáctico?

En las siguientes secciones se propondrán soluciones a estas preguntas. De esta manera, los aportes fundamentales del trabajo presentado son: Primero, una nueva forma para hacer el análisis sintáctico de los Lenguajes Visuales y segundo, la presentación de un caso real de utilización del mecanismo propuesto.

El presente artículo tratará el siguiende orden de ideas: 1. Introducción del artículo, 2. Presentación de la manera de comprobación sintáctica planteada, 3. Exhibición de un ejemplo en el que se usa la nueva forma de análisis sintáctico, 4. Una descripción de la forma como se implementó el editor y 5. Discusión de las conclusiones de la investigación y el desarrollo.

2. Alternativa para comprobación sintáctica de Lenguajes Visuales

Independiente de la especificación gramatical que se utilice la comprobación sintáctica tiene dos niveles: Sintaxis independiente del contexto y Sintaxis dependiente del contexto.

[1] Un compilador se denomina $cruzado$ cuando recibe el lenguaje fuente F, corre sobre una máquina que ejecuta código I y produce un código objeto O para otra máquina diferente a la que se emplea para ejecutar el compilador. Lo anterior se abrevia con la expresión $F_I O$ y se dibuja como: 

.

[2] Se denominan compiladores $escalonados$ aquellos procesadores de lenguajes que toman un par de compiladores $cruzados$ para obtener otro compilador con características diferentes.

[3] Combinación ordenada de significantes que interactúan formando un todo con sentido, dentro de un conjunto de reglas y convenciones sintácticas.

Según la jerarquía de Noam Chomsky, para especificar la sintaxis independiente de contexto de un lenguaje textual, se requiere a lo sumo, una gramática tipo 2, mientras que para especificar la sintaxis dependiente de contexto se necesita mínimo, una gramática tipo 1. De lo anterior, se puede concluir que desde el punto de vista computacional es más costosa la comprobación sintáctica dependiente del contexto que la independiente. Es por este motivo, que en la práctica se prefiere diseñar una gramática tipo 2 ó 3 (para tratar la sintaxis independiente de contexto y dejar el tratamiento de la dependiente en otras etapas posteriores), que construir una gramática tipo 1 (donde se contemplen todas las características posibles desde la sintaxis) y enfrentar un problema mucho más complicado. Esta metodología de dividir y conquistar sería muy útil en el tratamiento de Lenguajes Visuales si se pudieran abstraer y modelar las características independientes de contexto, dentro del análisis en una primera etapa desde el Lenguaje Visual y hacer que se conserven las características de sintáxis dependiente de contexto en el lenguaje textual de almacenamiento (de los programas inicialmente visuales), para su posterior análisis y tratamiento. Lo anterior, requiere que desde la gramática podamos obtener las características independientes de contexto, para esto se emplearán las Gramáticas de Sistemas de Íconos Generalizados, G_sig.

2.1. Introducción a las Gsig (Gramáticas de Sistemas de Íconos Generalizados)

Dentro de los principales problemas que se deben enfrentar al trabajar con lenguajes visuales se encuentra el establecimiento de una especificación gramatical, para así, determinar un mecanismo de traducción a lenguaje objeto y un mecanismo de almacenamiento de código intermedio. En otras palabras, requerimos guardar en algún medio los programas visuales siguiendo las reglas de alguna especificación gramatical, para posteriormente, mediante un mecanismo de traducción, obtener un código objeto.

En [3] se presenta la forma de especificación gramatical denominada Gsig, la cual consiste en la representación textual de la información visual de un determinado constructor o conjunto de constructores. Ampliando la idea, una Gsig almacena en una expresión textual, la información visual de cada constructor independiente X (referida como la parte física)[4], las relaciones con otros constructores del lenguaje (sintagmas llamados la parte lógica) y su respectiva etiqueta, organizadas como se presenta en la ecuación 1.

$$\mathcal{E}[\![X]\!]^{a} = S^{b}\ Sd_1{}^{c}\ \overbrace{(x-y)\ Path(X)}^{Parte\ Física}\ \sim^{d}\ \overbrace{Name(X)}^{Etiqueta}\ :^{e}\ \overbrace{\mathcal{E}[\![Ex(X)]\!]}^{Parte\ Lógica}\ Sd_2{}^{c} \tag{1}$$

[a] Función semántica que nos entrega la especificación del constructor visual X
[b] Símbolo de identificación que permite determinar la especificación de manera única.
[c] Componentes léxicos de delimitación de la especificación
[d] Símbolo de terminación de la Parte Física.
[e] Símbolo de terminación de la Etiqueta.

De forma similar, para el caso de la especificación de un conjunto de constructores visuales, una Gsig almacena en una expresión textual, la especificación de cada uno de los constructores y los correspondientes símbolos de sincronización $\widehat{Sc_i}$, como se muestra en la ecuación 2.

$$\mathcal{E}[\![X_1, \cdots, X_n]\!] = Sd_1\ \mathcal{E}[\![X_1]\!]\ \widehat{Sc_1} \cdots \widehat{Sc_{n-1}}\ \mathcal{E}[\![X_n]\!]\ Sd_2 \tag{2}$$

2.2. Definición formal del nuevo mecanismo de análisis sintáctico para Lenguajes Visuales

Antes de continuar se necesita una definición.

[4] Partes que componen los íconos generalizados presentados por Chang en [1]

Definición 1. *El conjunto* **PRIMEROS_ÍCONOS** *consta de los primeros íconos que se puedan generar de una producción.*

$$\textit{PRIMEROS\_ÍCONOS}(X) = \begin{cases} imagen_X & \textit{Si } X \rightarrow S'\ Sd_1\ (X - Y)\ imagen_X\ \sim\ \ldots, \\ \textit{PRIMEROS\_ÍCONOS}(PL_X{}^a) & \textit{Si } X \rightarrow S'\ Sd_1\ \epsilon\ \sim\ Et_X{}^b\ :\ PL_X\ Sd_2, \\ \bigcup \textit{PRIMEROS\_ÍCONOS}(X_i) & \textit{Si } X \rightarrow S'\ Sd_1\ X_1\ \widehat{Sc_1}\ \cdots\ \widehat{Sc_{n-1}}\ X_n\ Sd_2, \\ \textit{PRIMEROS\_ÍCONOS}(Y \veebar Z)^c & \textit{Si } X \rightarrow Y \mid Z. \end{cases} \tag{3}$$

[a] Parte Lógica del constructor X
[b] Etiqueta dada por el usuario al constructor X
[c] O exclusivo entre las producciones Y y Z

Ya que las Gramáticas de Sistemas de Íconos Generalizados tienen una estructura que las clasifica dentro de la jerarquía de Noam Chomsky como de tipo 2, podemos afirmar que cada producción nos determina unas características independientes del contexto así:

- Características independientes de contexto en las producciones de los operadores independientes:

$$X \rightarrow S'\ Sd_1\ ParteFísica_X\ \sim\ Etiqueta_X\ :\ \underbrace{ParteLógica_X}_{LX}\ Sd_2 \tag{4}$$

  1. Los símbolos de sincronización: $S',\ Sd_1,\ \sim$ y **:** . Están determinados para cada constructor independiente del lenguaje.
  2. La $ParteFísica$ de cada constructor está determinada por la aplicación de edición de programas visuales.
  3. La ventana resultado de la $expansión$ del ícono etiquetado con $Etiqueta_X$ sólo podrá contener íconos que pertenezcan a PRIMEROS_ÍCONOS(LX).

- Características independientes de contexto en las producciones de los conjuntos de constructores dentro de una $expansión$:

$$LX \rightarrow S'\ Sd_1\ X_1\ \widehat{Sc_1}\ \cdots\ \widehat{Sc_{n-1}}\ X_n\ Sd_2 \tag{5}$$

  1. Los símbolos de sincronización: $S'\ Sd_1,\ Sd_2\ \widehat{Sc_1}$ ... $\widehat{Sc_{n-1}}$. Están determinados para cada conjunto de constructores dentro de una $expansión$.
  2. La ventana LX únicamente puede contener íconos que pertenezcan a PRIMEROS_ÍCONOS(LX).

Se necesita un par de definiciones para proseguir.

Definición 2. *En adelante, llamaremos* **Ventana de Expansión** *al área de trabajo resultande de* $expandir$ *un ícono en alguna aplicación para edición de programas visuales y donde se pueden desplazar íconos de contructores del lenguaje. De igual forma, una Ventana de Expansión* **Activa** *es una Ventana de Expansión que tiene el enfoque en ese instante, se representa por una ventana con los bordes más gruesos (ver figura 2).*

Definición 3. *En lo subsiguiente se hará referencia como* **Ventana de Constructores Activos** *al repositorio de íconos de constructores del lenguaje (presente en una aplicación para la edición de programas visuales) que pueden ser desplazados hacia su respectiva Ventana de expansión activa para su utilización. Se muestra como una ventana con los íconos organizados en vertical (ver figura 2).*

Consecuente con las definiciones 2 y 3 se puede definir qué constructores pueden intervenir en la parte lógica de cada constructor visual expresado con una Gsig.

Definición 4. *Los* **Íconos Activos** *son las gráficas que aparecerán en una ventana de constructores activos y que podrán ser desplazados hacia una ventana de expansión determinada.*

Si X es un constructor visual, entonces en su Ventana de Expansión (cuando esté Activa) sólo se podrán desplazar los respectivos íconos activos, los cuales corresponden a PRIMEROS_ÍCONOS(LX).

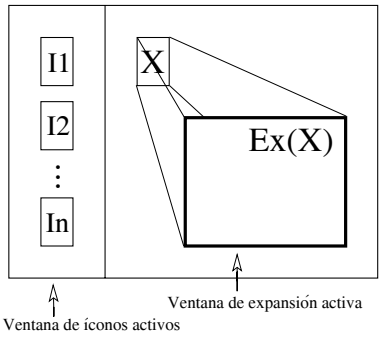


Figura 2: Ventana de constructores activos y Ventana de expansión activa.

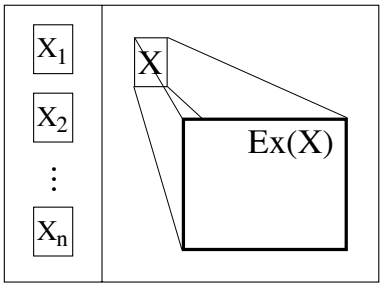


Figura 3: Íconos permitidos en la ventana de expansión activa del ícono del constructor X.

$$X \rightarrow S'\ Sd_1\ ParteFísica \sim Etiqueta_X : \underbrace{ParteLógica}_{LX}\ Sd_2$$

$$LX \rightarrow S'\ Sd_1\ X_1\ \widehat{Sc_1}\ \cdots\ \widehat{Sc_{n-1}}\ X_n\ Sd_2$$

$$X_1 \rightarrow S'\ Sd_1\ (x-y)\ imagen_{X_1} \sim \cdots$$
$$\vdots$$
$$X_n \rightarrow S'\ Sd_1\ (x-y)\ imagen_{X_n} \sim \cdots$$

$$\textit{PRIMEROS\_ÍCONOS}(LX) = \{imagen_1,\ \ldots,\ imagen_n\}$$

$$imagen_{X_i} \stackrel{\text{def}}{=} \boxed{\text{Xi}}$$

De esta manera, se puede restringir la utilización del conjunto de íconos en cada ventana del editor de programas visuales asegurando que en cada ventana de expansión (parte lógica del constructor respectivo) solamente se permitan desplazar los íconos de constructores adecuados, logrando de esta forma, que todo programa editado en un entorno que contenga y emplee la definición de las ventanas de constructores activos será gramaticálmente correcto con respecto a su sintaxis independiente del contexto. Pudiendo así, generar el programa en código textual de almacenamiento sin errores sintácticos independientes del contexto, para analizarlo en función de la sintaxis dependiente del contexto.

De esta forma, se ha dividido el análisis sintáctico de un lenguaje visual en dos etapas diferentes:

1. Análisis sintáctico independiente de contexto: que se efectúa al momento de la edición del programa visual no permitiendo que el usuario pueda desplazar un ícono a una ventana que no le corresponde o, en otras pala-

bras, se restringen los íconos activos dependiendo de la ventana de expansión activa, en términos de lingüística computacional, se acota el conjunto de lexemas que pueden aparecer en un contexto (se define un paradigma[5]).

2. Análisis sintáctico sensible al contexto: Se efectúa (en las etapas de análisis semántico y traducción) sobre el modelo textual de almacenamiento con la precondición de que está correcto con respecto a la sintáxis independiente del contexto. Lo anterior, se consigue empleando un mecanismo de traducción como el T_Gsig [4], además de la inclusión de tablas de símbolos, (para un recorrido eficiente se emplean Tablas Hash) en donde se almacena información de: identificadores (ámbito, clase de identificador), declaraciones de ámbito (lista de identificadores de variables y métodos), envío de mensajes (lista de identificadores), métodos (lista de parámetros), para recorrer cada una de las tablas y verificar las condiciones dependientes del contexto como: todas las variables debe tener un valor asociado, la cantidad de argumentos en un método deben ser igual a la cantidad de parámetros declarados, se deben emplear métodos existentes. Lo anterior, a sabiendas de que el código recibido cumple con las características sintácticas independientes del contexto.

3. Ejemplo: El editor del Cálculo Visual GraPiCO, E_GraPiCO

Ante el requerimiento de una aplicación que permita la programación en Calculo GraPiCO [5] del Grupo de Invetigación AVISPA[6] se diseñó e implementó un editor visual denomidado E_GraPiCO; para efectuar su parser se empleó el mecanismo de comprobación sintáctica propuesto en este artículo. En las secciones posteriores se hace su introducción.

3.1. Alfabeto de GraPiCO

El la figura 4 se ilustran los íconos que pueden a parecer en un momento dado dentro de un programa GraPiCO (Ventana de Íconos Activos más general) y por medio de glosas se explican cada uno de los elementos gramaticales utilizados en E_GraPiCO, esta también es una forma de presentar el grupo de Íconos Activos de cada constructor que genere una Ventana de Expansión

3.2. Descripción del área de trabajo (Características del editor)

Los lenguajes visuales están basados en un lenguaje icónico que representa los elementos del problema en discusión. Por tanto se propuso crear una interfaz que permita construir programas gráficamente mediante un alfabeto gráfico definido, facilitándole al programador representar en un área de diseño las diferentes instrucciones GraPiCO.

Para ello, la interfaz de desarrollo ofrecerá un entorno amigable para el programador, con las respectivas herramientas gráficas necesarias para la total representación de las sentencias del cálculo GraPiCO. Las herramientas deben contener las características indispensables para la comodidad del usuario durante el desarrollo de sus programas; entre estas características encontramos:

- La validación sintáctica definiendo la disponibilidad de las herramientas según el tipo de sentencia que se esté programando.
- El control de las ayudas contextuales y elementos gráficos de soporte al usuario.
- La navegabilidad entre ventanas para describir la estructura del programa GraPiCO.
- Barras de herramientas, que permiten ejecutar funcionalidades muy repetidamente.

3.3. Representación de un Programa GraPiCO

Se desarrolló un entorno visual que le brinda al programador las herramientas gráficas necesarias para diagramar sobre una Lista de Ambientes (área de trabajo), obteniendo como resultado un Programa GraPiCO. Un Programa GraPiCO se compone de una Lista de Ambientes y a su vez cada Lista de Ambientes puede tener una Lista de Constructores; un Ambiente es el espacio donde el programador dibujará las instrucciones que constituyen su Programa (formalmente una Ventana de Expansión). Donde cada Ambiente utilizado tiene definido un Conjunto de Gráficos que se relacionan entre sí para expresar una instrucción, y a su vez son el punto de partida para generar nuevos Ambientes.

[5] Conjunto virtual de elementos de una misma clase gramatical, que pueden aparecer en un mismo contexto.

[6] **A**mbientes **Vis**uales de **P**rogramación **A**plicativa, Universidad Javeriana, Universidad Del Valle, Universidad De Los Andres, Instituto IRCAM

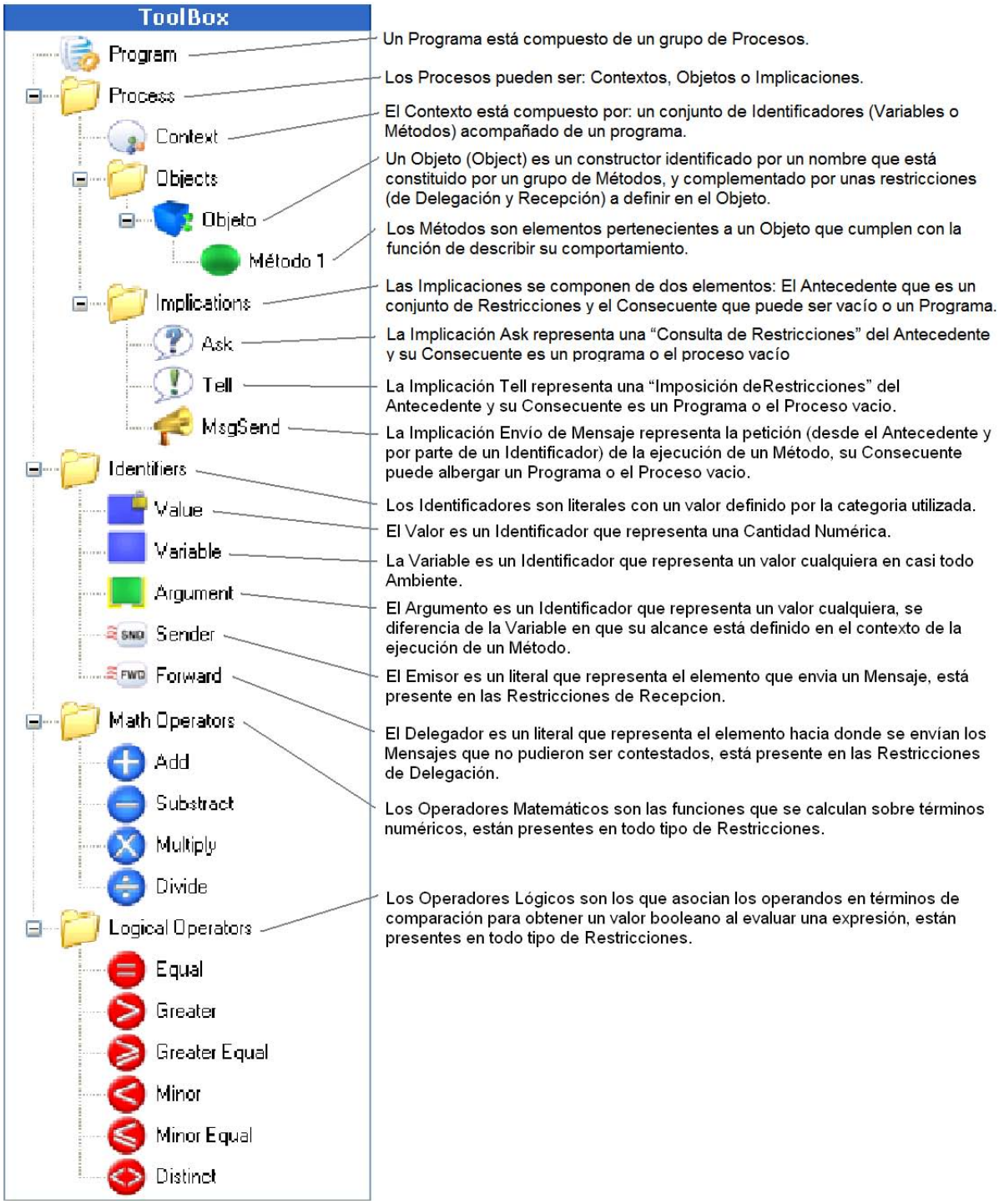


Figura 4: Representación del lenguaje icónico manejado a partir de la gramática del Cálculo Visual GraPiCO.

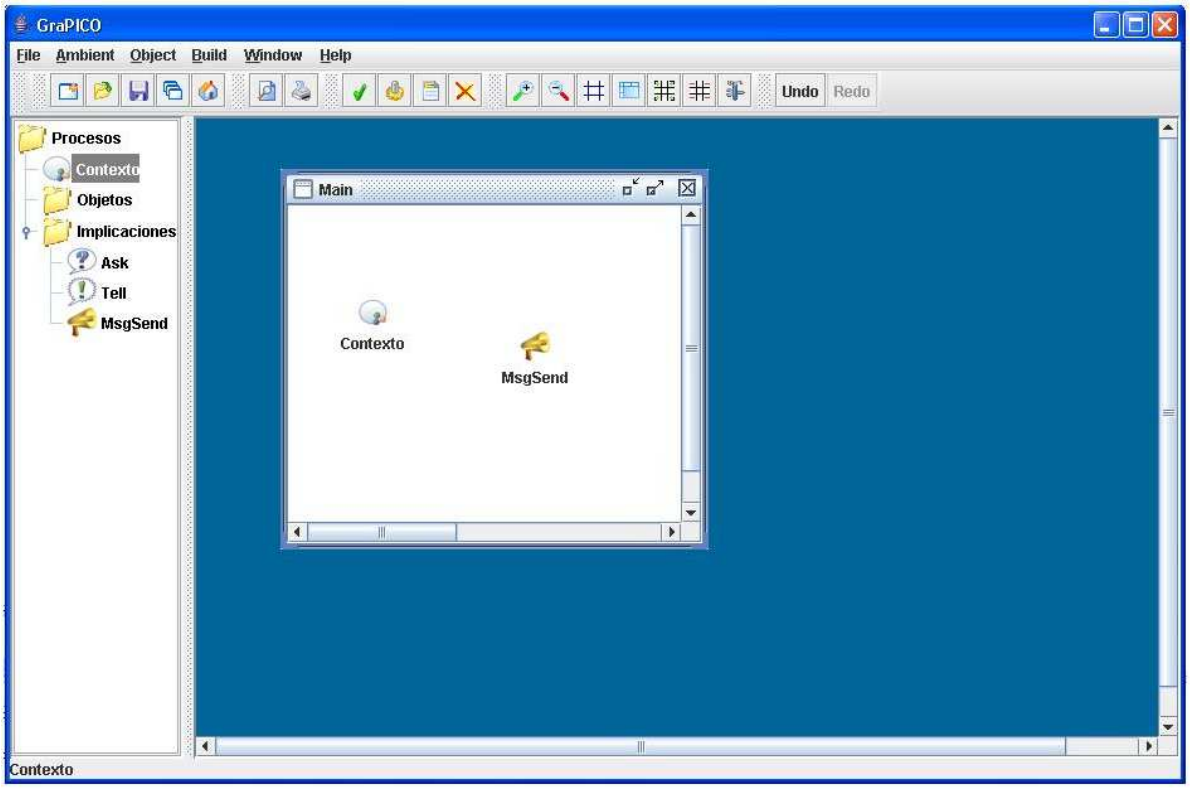


Figura 5: Diseño del formulario Principal

Los objetos definidos pueden ser usados como procesos en distintos Ambientes dependiendo de las Restricciones Sintácticas visuales impuestas[7].

3.4. Componentes del entorno gráfico

El entorno debe proporcionar los mecanismos necesarios para que el programador dibuje de manera cómoda las instrucciones de su Programa, utilizando las herramientas gráficas, sobre los Ambientes o Paneles. Los mecanismos ofrecidos por E_GraPiCO son:

- Menú: Despliega un listado de las opciones disponibles del entorno gráfico de acuerdo a las categorías en que las opciones se encuentren agrupadas.
- Barra de Herramientas: Despliega una serie de íconos que corresponden a algunas herramientas utilizadas para tareas que proporciona el entorno, las cuales son de uso frecuente por el programador de Cálculo Visual GraPiCO.
- Caja de herramientas (Ventana de Íconos Activos): Despliega organizadas en un árbol de jerarquías, las herramientas disponibles para la construcción de las sentencias del Cálculo Visual GraPiCO, dependiendo de las Restricciones de cada Ambiente.
- Área de construcción del programa: En esta área serán organizados los paneles de diseño para la construcción del programa. Esta área es la que contendrá todas las ventanas, donde cada ventana representa un Ambiente y éste a su vez una instrucción GraPiCO.
- Formulario principal: Es la ventana que integra todos los elementos mencionados anteriormente.

[7] Definición de los conjuntos **Íconos Activos** para cada Ambiente

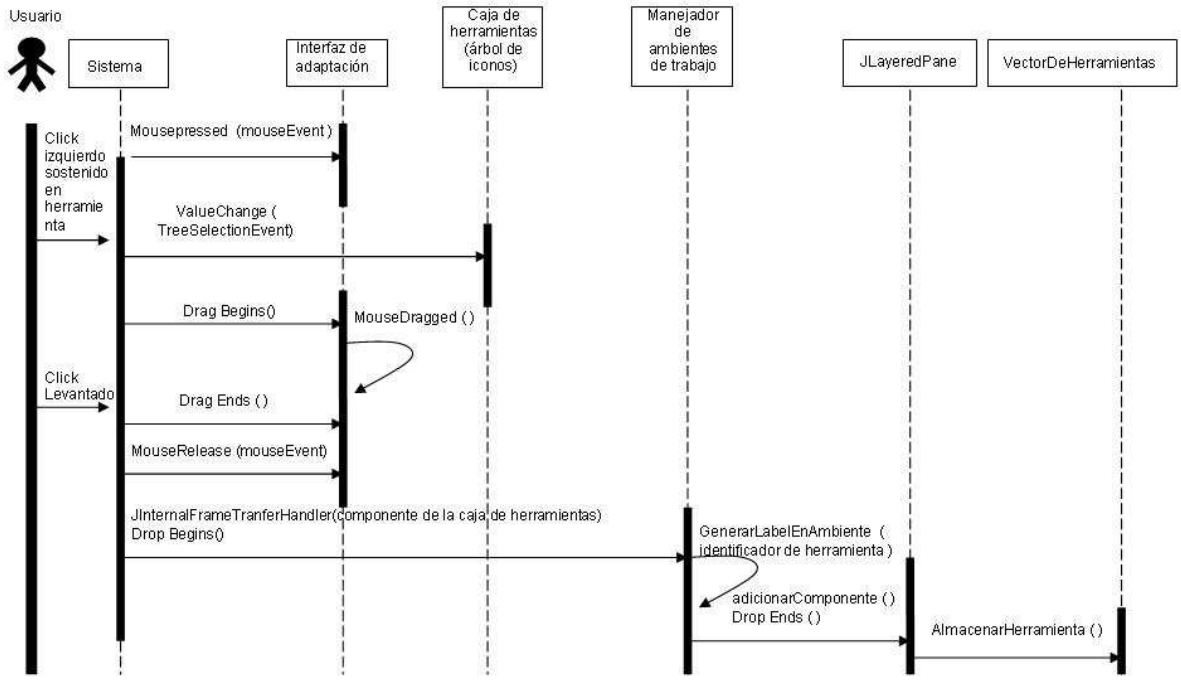


Figura 6: Diagrama de secuencia DnD.

Para dibujar una sentencia en un programa GraPiCO, el programador debe seleccionar de la caja de herramientas un ícono que representa un elemento de la gramática utilizada, arrastrarlo y soltarlo sobre el Ambiente o ventana actual del área de construcción. Cuando un ícono es soltado sobre un Ambiente se debe interpretar sus características y dibujarlo sobre el Ambiente en donde indique la posición del puntero del Mouse.

Para mostrar la manera como se comporta la aplicación se presentarán los diagramas de secuencias más relevantes en los apartados siguientes.

3.5. Diagrama de secuencia Drag And Drop (DnD) de la Caja de Herramientas hacia los ambientes de trabajo

El diagrama de la figura 6 presenta el manejo del desplazamiento de los constructores de la caja de herramientas (Ventana de Íconos Activos) hacia el ambiente (Ventana de Expansión Activa).

3.6. Diagrama de secuencia de activación de un Ambiente y actualización de la Caja de Herramientas

La figura 7 contiene el diagrama que muestra como se efectúan los cambios en la caja de herramientas (conjunto de Constructores que aparecen en la Ventana de Iconos Activos) dependiendo del ambiente (Ventana de Expansión) activo.

4. Descripción del Desarrollo del editor

4.1. Herramientas para el desarrollo del proyecto

Como herramienta, en el lenguaje de programación JAVA, existe el entorno de desarrollo NETBEANS IDE basado en el software de NETBEANS.org que proporciona mecanismos para escribir, diseñar, compilar, depurar y ejecutar programas JAVA ayudando en la generación de código, documentación (Developer collaboration) y de clases que hacen énfasis en el manejo de API, como por ejemplo, Swing (Grupo de componentes escritos en java para diseñar interfases graficas de usuario multiplataforma), JAVA 2D (incorporación de gráficos 2D de alta calidad, texto e imágenes) y, soporte Drag and Drop (arrastrar y soltar: trasferencia de objetos gráficos entre paneles y aplicaciones).

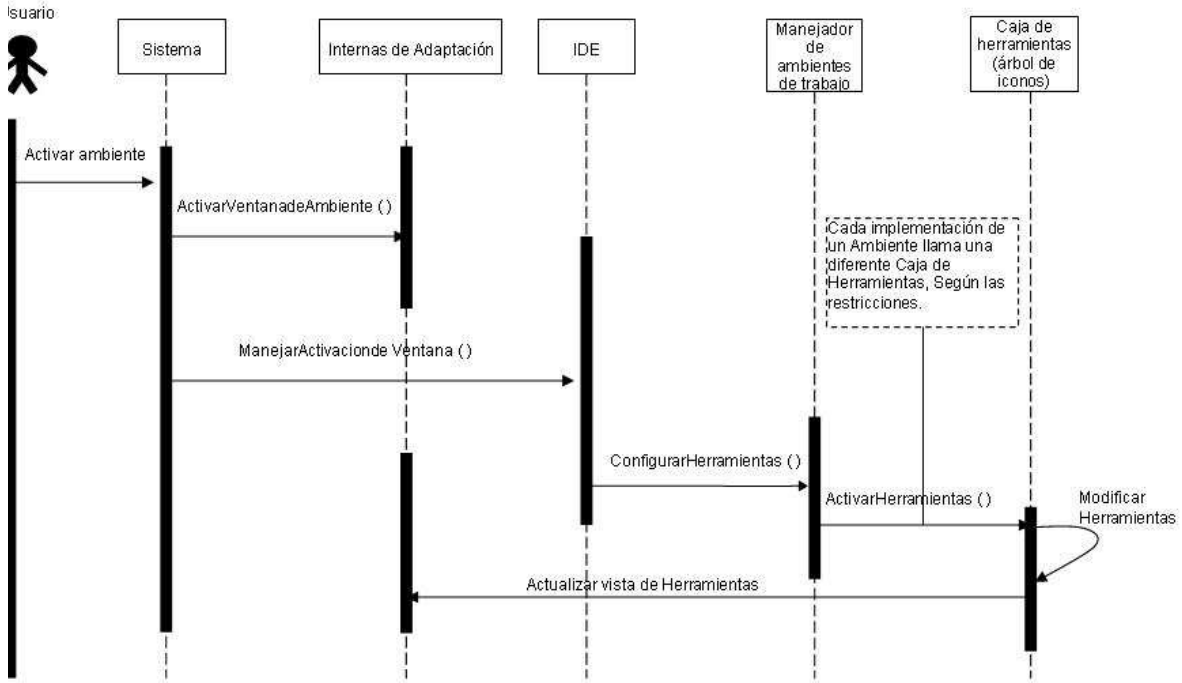


Figura 7: Diagrama de secuencia de activación y actualización.

4.2. Funcionalidades del Editor Gráfico para El Cálculo Visual GraPiCO.

4.2.1. Análisis Sintáctico Dinámico

Uno de los avances del editor, es la propiedad de realizar un análisis sintáctico dinámico, cuya función es analizar la manera en que el usuario está utilizando el lenguaje icónico del Cálculo Visual GraPiCO y que concuerde con los términos de la gramática expuestos en [5], mientras se lleva a cabo conjuntamente la creación de él, concluyendo en que todas las listas de Ambientes generadas en el Programa hecho por el usuario, se ejecute sin errores sintácticos (de orden) a partir de reglas sintácticas creadas que convierten el programa visual con cada unos de sus elementos gráficos a una forma textual, con el objetivo de entregarle a las siguientes etapas de compilación un programa almacenado de forma textual sintácticamente correcto y que el compilador no tenga que realizar un análisis sintáctico. El Análisis Sintáctico Dinámico está implícito en el editor, haciéndolo dinámico debido a que el usuario no puede cometer errores de programación.

Para utilizar la teoría expuesta en la sección 2.2 la implementación del entorno gráfico de E_GraPiCO se diseñó de manera que la Ventana de Constructores Activos (planteada en la Definición 3) aparece representada por la caja de herramientas (ToolBox en el desarrollo), donde se situan los dibujos con los que se puede construir los programas GraPiCO (el conjunto *PRIMEROS_ÍCONOS(X)* introducido en la Definición 1) mediante su desplazamiento hacia las ventanas en el Área de Construcción de Programa o Desktop (estas ventanas son denominadas Ambients, tipo específico de Frame en la implementación para la construcción de sentencias de lenguaje); la conformación del grupo de íconos está condicionada por la ventana que en ese instante tenga el enfoque (mostrada como Ventana de Expansión Activa en la Definición 2), así, cuando un determinado ícono es expandido, su correspondiente conjunto de Íconos Activos (según la Definición 4) es dispuesto en la caja de herramientas para hacer posible su empleo en la edición de programas.

Para desarrollar este análisis es necesario lograr que cada ícono de constructor del lenguaje active su respectiva caja de herramientas. Esta relación entre constructores y conjunto de Íconos Activos en la caja de herramientas se puede modelar mediante una tabla de reglas de disponibilidad de herramientas para los Ambients como la que se utilizó para E_GraPiCO y que se presenta en la figura 9. Un ejemplo para demostrar esta funcionalidad, es cuando el usuario se encuentra en un Ambient Método, en donde sólo podrá desplazar de la caja de herramientas los constructores de Programa, Proceso e Identificadores. La operabilidad de estas reglas en el editor es exhibida en el diagrama de secuencia en la figura 8.

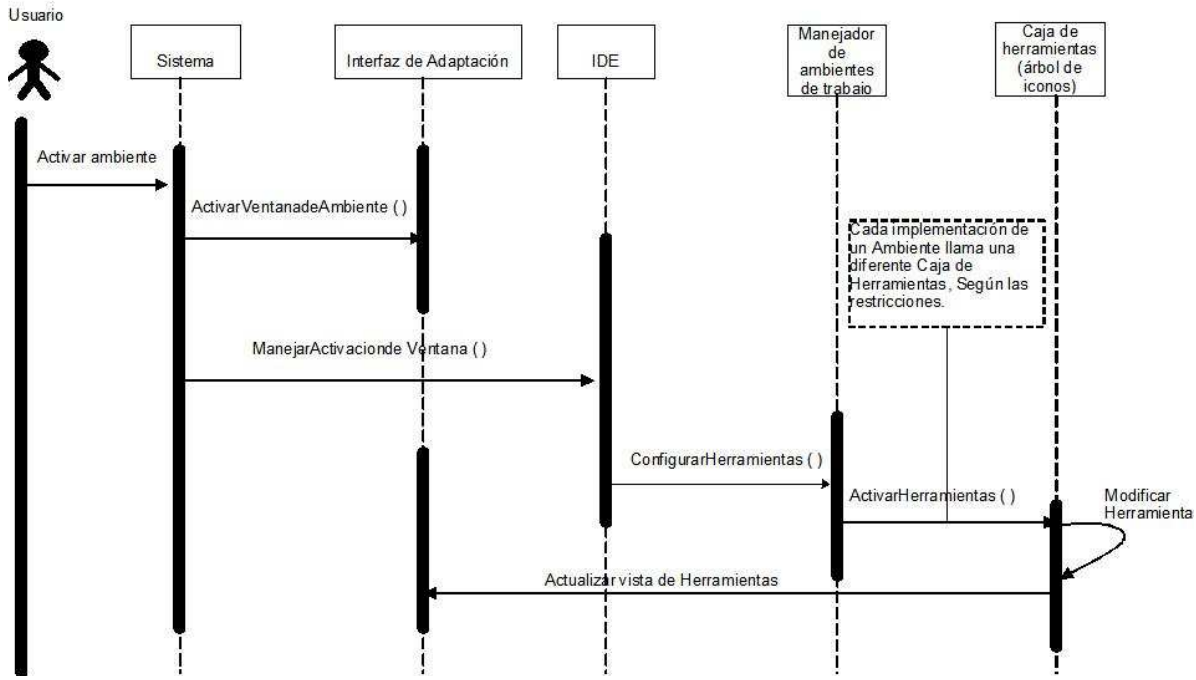


Figura 8: Diagrama de secuencia de actualización de Ambients y actualización de la caja de herramientas.

Constructor \ Ambients	Prog.	Context	Method	Constraints	Constraints Group	Msg Send
Program		X	X			
Process	X	X	X			
Method		X				X
Math Operators				X		
Logical Operators				X		
Identifiers		X	X	X		X
Constraints					X	

Figura 9: Tabla de la relación entre constructores y ventanas (Ambients)

4.2.2. Almacenamiento Textual de los Programas GraPiCO

Frente a la necesidad de brindar la posibilidad de guardar un programa para su posterior edición o ejecución se empleó una representación textual de cada constructor utilizado en el Cálculo Visual GraPiCO por medio de las G_sig presentadas en [3], de manera que el archivo almacenado conserve la jerarquía generada en los programas. Es decir, a partir del Ambient principal (conocido en el editor como Main), se crea un orden para el conjunto de contructores utilizados siguiendo el árbol sintáctico del programa.

5. Conclusion

De esta manera, mediante el mecanismo de $dividir\ y\ conquistar$ se ha reducido la dificultad del análisis sintáctico de un programa visual, pues en lugar de hacer el análisis sintático en su totalidad desde el programa visual, se plantea una primera etapa de análisis sintáctico independiente de contexto de manera dinámica, y se efectúa el análisis dependiente de contexto de manera estática, es decir, se abstraen las comprobaciones sintáticas que se pueden hacer al momento de la edición (sin mucho costo computacional) de las que requieren mayor tratamiento y almacenamiento de información, de otra forma, el análisis dependiente de contexto.

Gracias a los métodos formales que se emplearon para presentar el nuevo mecanismo es posible la demostración de correción de la etapa de análisis sintáctico a un lenguaje de programación.

por medio de la implementación de E_GraPiCO se pudo constatar que, en efecto, la nueva forma de análisis sintactico introducida es útil y se puede ayudar en la elaboración de nuevos lenguajes de programación visual donde se requiera un parsing agil y correcto.

Referencias

[1] CHANG, S. In *Principles of Visual Programming Systems.* (1990).

[2] COSTAGLIOLA, G., AND POLESE, G. Extended positional grammars. In *Tecnical report, Dipatimento di Matematica ed Informatica, University of Salerno, Salerno, Italy* (2001).

[3] TAVERA, C., AND DÍAZ, J. Alternativa para especificación sintáctica: Gramática de sistemas de íconos generalizados: Gsig. In *Congreso Argentino de Ciencias de la Computación, San Luis, Argentina.* (2006).

[4] TAVERA, C., AND DÍAZ, J. Alternativa de mecanismo de traducción para lenguajes visuales: T_gsig. In *Reporte técnico; http://eisc.univalle.edu.co/∼catavera* (2007).

[5] TAVERA, C., AND DÍAZ, J. Nuevo cálculo visual: Grapico. In *Congreso Colombiano de Computación, Universidad Javeriana, Bogotá, Colombia* (2007).

Análisis de herramientas MDA

Verónica A. Bollati, Juan M. Vara, Belén Vela y Esperanza Marcos
Grupo Kybele
Universidad Rey Juan Carlos
C/ Tulipán S/N, 28933 - Móstoles (MADRID)
{veronica.bollati, juanmanuel.vara, belen.vela, esperanza.marcos}@urjc.es

Abstract

Nowadays, the MDA proposal is becoming more and more important in software development. In the last years, a lot of tools that support MDA have appeared to make easier the system development task. Following the MDA guidelines, MIDAS, a model architecture for the development of information systems, that proposes metamodels based on UML profiles, has appeared. In this paper, a study of two MDA tools (AndroMDA and ArgoUML) and a development framework (Eclipse) is presented, in order to decide on the need of the development of a new tool, or the adaptation of one of the existing ones, for giving support to MIDAS. For this comparative study, at first, a characterization of the MDA tools have been made, that is, the set of characteristics that, in our opinion, are mandatory or desirable in an MDA tool. Afterwards, these characteristics have been analysed in the selected tools, by means of applying them to a case study, allowing in this way the evaluation of their functionality. The obtained conclusions will allow us to decide which technology to use and which will be the MDA architecture to give support to MIDAS.

Keywords: Software Engineering, MDA, MDA Tools, MIDAS, metamodel, UML, profile

Resumen

En la actualidad la propuesta MDA está tomando cada vez más fuerza en el desarrollo de software. En los últimos años han ido apareciendo numerosas herramientas que dan soporte a esta propuesta, facilitando la tarea del desarrollador de sistemas de información. Siguiendo las líneas de MDA nace MIDAS, que es una arquitectura de modelos para el desarrollo de sistemas de información, que propone meta-modelos basados en perfiles UML. En este artículo se presenta un estudio comparativo sobre herramientas MDA (AndroMDA y ArgoUML) y entornos de desarrollo (Eclipse), con el objetivo de determinar la necesidad de desarrollo de una nueva herramienta o la adaptación de alguna de las existentes para dar soporte a MIDAS. Para realizar dicha comparativa se ha realizado en primer lugar una caracterización de herramientas MDA; es decir, se han determinado un conjunto de características que, en nuestra opinión, son obligatorias o deseables en una herramienta MDA. Posteriormente, se han analizado dichas características en las herramientas seleccionadas, mediante su aplicación a un caso de estudio, permitiendo de esta forma evaluar la funcionalidad de las mismas. Las conclusiones obtenidas en este estudio permitirán decidir la tecnología a usar y la arquitectura de la herramienta MDA que dará soporte a MIDAS.

Palabras Claves: Ingeniería de Software, MDA, Herramientas MDA, MIDAS, meta-modelo, UML, perfil

1 INTRODUCCIÓN

MDA [16], propuesto por el grupo OMG, está tomando cada vez más fuerza en el desarrollo de software. Se trata de un marco de trabajo para el desarrollo de software, cuya principal característica es la definición de modelos como elementos de primer orden en el diseño, desarrollo e implementación del software y las transformaciones entre los diferentes modelos mediante la definición de *mappings*. MDA considera diferentes tipos de modelos, en función del nivel de abstracción de los mismos: los requisitos del sistema son detallados en el Modelo Independiente de Computación (*Computation Independent Model*, CIM); en los Modelos Independientes de Plataforma (*Platform Independent Model*, PIM) se representa la funcionalidad del sistema sin considerar la plataforma final y los Modelos Específicos de Plataforma (*Platform Specific Model*, PSM) se obtienen de combinar las especificaciones contenidas en el PIM con los detalles de la plataforma elegida. A

partir de los diferentes PSMs se pueden generar automáticamente distintas implementaciones (código) del mismo sistema.

En los últimos años han aparecido numerosas herramientas MDA que permiten en mayor o menor grado automatizar las transformaciones y que generan, en algunos casos, (semi-) automáticamente código para distintas plataformas.

MIDAS [11][21] es una arquitectura de modelos para el desarrollo de Sistemas de Información (SI) basado en MDA, que propone modelar los sistemas de acuerdo a dos dimensiones ortogonales: el grado de dependencia de la plataforma (CIM, PIM y PSM) y los aspectos, en los que comúnmente se estructura el SI (Web) (contenido, hipertexto y comportamiento).

Con el fin de dar soporte a los modelos y las transformaciones definidas en MIDAS se pretende desarrollar una herramienta MDA. Para ello se realizará una revisión de algunas de las herramientas MDA existentes para determinar:

- Las características *obligatorias* y las *deseables* en una herramienta MDA.
- Las características que recogen la mayoría de las herramientas.
- Y las principales carencias que presentan las mismas.

Existen otros trabajos relacionados con la evaluación de herramientas MDA, como [6][8][15][19]. Tras su análisis, en el presente nos hemos basado en [6] y [19]. En [6] se realiza un estudio comparativo de dos herramientas MDA comerciales (OptimalJ y ArcStyler). En dicho estudio se han evaluado las herramientas realizando un análisis de propiedades extraídas de la especificación de MDA y desarrollando para ello un caso de estudio, con el fin de detectar cuál de las dos se ajusta mejor a MDA. En [19] se presenta un marco de referencia para la evaluación de herramientas MDA y se evalúan diez herramientas de estas, logrando realizar una taxonomía de las mismas. Para ello, se propone, una clasificación de herramientas, así como los criterios de evaluación. Estos últimos se basan en características de MDA, criterios de calidad, de entorno y generales.

En el trabajo que aquí se presenta se ha realizado un compendio de las características propuestas en los dos trabajos anteriormente citados, seleccionando las características que se consideran obligatorias o deseables que cumpla una herramienta MDA. Además se han introducido nuevas características a ser evaluadas. Para la selección de las herramientas se ha considerado, por un lado, la propuesta de clasificación de herramientas del segundo trabajo [19] y por otro lado, la conveniencia de abarcar tanto herramientas comerciales como herramientas de libre distribución; teniendo en cuenta en todo momento que es necesario que las herramientas seleccionadas sean extensibles, ya que esto nos permitirá dar soporte a MIDAS. El objetivo principal de este trabajo es evaluar las herramientas para verificar si es posible adaptar alguna de ellas o, si por el contrario, es necesario desarrollar una nueva herramienta para dar soporte a MIDAS.

Este artículo se estructura de la siguiente manera: en el apartado 2 se realiza una caracterización de las herramientas MDA; en el apartado 3 se presenta el caso de estudio que se utilizará para la evaluación; en el siguiente apartado se presenta una breve descripción de las herramientas seleccionadas y el resultado de la implementación del caso de estudio con las mismas; la sección 5 es una discusión que incluye una comparativa de las herramientas analizadas; finalmente, en el apartado 6 se recogen las principales conclusiones y se plantean trabajos futuros.

2 CARACTERIZACIÓN DE HERRAMIENTAS MDA

La evolución de las herramientas de apoyo al proceso de desarrollo de SI está ligada a la evolución

de la Ingeniería de Software como disciplina. Las primeras herramientas fueron los editores y procesadores de texto, así como las herramientas de dibujo que incorporaban notaciones gráficas. A partir de la consolidación de las metodologías de desarrollo, integrando diferentes técnicas, comenzaron a aparecer las primeras herramientas CASE (*Computer–Aided Software Engineering*), seguidas posteriormente por las herramientas extensibles, herramientas de transformación y herramientas generadoras de código. Por ultimo, se pueden citar las *herramientas MDA,* en las que se centra el presente artículo. Su principal aportación es dar soporte a la transformación de modelos, permitiendo realizar, en la mayoría de los casos, transformaciones verticales y horizontales [20]. Las transformaciones se deben realizar de forma (semi-)automática a través de la definición de reglas de transformación o *mappings*.

Para la selección de las características que se han estudiado en las diferentes herramientas, se han tenido en cuenta las propuestas en los trabajos citados anteriormente, completándose los mismos con otras características extraídas de la especificación de MDA. Éstas se han agrupado en funcionales, técnicas y de calidad, indicándose para cada una de las características el grado de cumplimiento que debería tener la herramienta de acuerdo a las necesidades de MIDAS: obligatoria (*Obl*) o deseable (*Des*).

2.1 Características Funcionales

En esta categoría se agrupan todos aquellos requisitos que se correspondan con los aspectos funcionales que debería cumplir una herramienta MDA:

- **Niveles que cubre (Obl):** qué niveles de MDA cubre (CIM, PIM, PSM).
- **Grado de generación de código (Obl):** en qué medida permite la transformación de los PSMs a código. Por ello se debe evaluar si las herramientas generan código y si dicho código puede ser implementado de forma directa o si es necesario realizar modificaciones en el mismo. Además se debe verificar si la generación de código se realiza desde el nivel PSM o desde el nivel PIM.
- **Transformaciones (Obl):** verificar el grado de automatización de las transformaciones.
- **Interacción con el usuario (Des):** verificar el grado de participación del usuario en el proceso de transformación.
- **Tipo de transformaciones (Obl):** evaluar, si permite realizar transformaciones verticales y/u horizontales. Se entiende por transformaciones verticales, las que se realizan entre los diferentes niveles de abstracción (CIM, PIM y PSM) y, por horizontales, las que se realizan entre los diferentes modelos de un mismo nivel de abstracción.

2.2 Características Técnicas

En esta categoría se agrupan todos aquellos requisitos que se correspondan con las características técnicas que debería cumplir una herramienta MDA:

- **Lenguaje de almacenamiento y gestión de modelos (Obl):** para realizar las transformaciones de los modelos entre los distintos niveles de abstracción es necesario un lenguaje que permita el almacenamiento y la gestión de estos modelos de forma que éstos se puedan intercambiar entre los distintos niveles (CIM, PIM y PSM). Es importante que el lenguaje utilizado se base en algún estándar, para asegurar la integración entre herramientas.
- **Plataformas y tecnologías soportadas (Des):** verificar qué plataformas de desarrollo soportan.
- **Ámbito de aplicación (Des):** se debe determinar el tipo de desarrollo en el que se centran, en particular para el presente artículo interesa determinar si las herramientas están orientadas hacia el desarrollo de SI Web o hacía desarrollos orientados a servicios, o ambos.

2.3 Características de Calidad

En esta categoría se agrupan todos aquellos requisitos que se correspondan con las características de calidad que debería cumplir una herramienta MDA:

- **Uso de Estándares (Des):** es importante determinar si las herramientas estudiadas hacen uso de estándares como UML, XML y MOF, para la definición de los modelos. Además se tendrá en cuenta si permite definir o personalizar perfiles propios del usuario para UML.
- **Extensibilidad (Obl):** capacidad que tiene la herramienta de adaptarse a nuevos requisitos.
- **Usabilidad (Des):** determinar la medida en el que la herramienta puede ser usada por los usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado.
- **Interoperabilidad entre herramientas (Des):** ver en qué grado la herramienta se puede integrar con otras herramientas con el objetivo de obtener nuevas funcionalidades.

3 CASO DE ESTUDIO

Con el fin de evaluar las distintas herramientas seleccionadas, se ha usado como caso de estudio el perfil UML para el meta-modelo Objeto-Relacional (OR) [12] [21] del estándar SQL:2003 [7]. Como se ha dicho anteriormente, MIDAS contempla varios aspectos: contenido, hipertexto y comportamiento. En el aspecto de contenido, que es en el que se centra el presente artículo, propone dos meta-modelos para la implementación del nivel PSM, el meta-modelo del *XML Schema* y el meta-modelo OR. Este último es el que se ha utilizado para el desarrollo del caso de estudio.

El perfil OR permitirá la especificación de modelos de datos OR, mediante extensiones de UML, para cada una de las herramientas elegidas. Posteriormente, para realizar la validación de este meta-modelo, se creará una base de datos para una empresa de arquitectura utilizando para ello, el perfil OR previamente creado.

Con este caso de estudio se pretende determinar el grado de cumplimiento de las herramientas para cada una de las características detalladas anteriormente.

3.1 Meta-modelo objeto relacional

El meta-modelo OR para el estándar SQL:2003, que aquí se presenta, recoge sólo las extensiones para objetos del estándar, no se incluyen los artefactos correspondientes al meta-modelo relacional. Los tipos de datos se clasifican en *Array, Multiset, Row*, tipo referencia y tipos estructurados. Una tabla tipada está basada en un tipo estructurado. Un tipo estructurado puede contener atributos y métodos. Una descripción detallada de dicho meta-modelo puede encontrarse en [20].

En la Figura 1 se muestra el perfil UML [4] para el meta-modelo OR del estándar SQL:2003. Dicho perfil permitirá el modelado de un esquema OR (nivel PSM de la arquitectura de MIDAS) usando un diagrama de clases extendido.

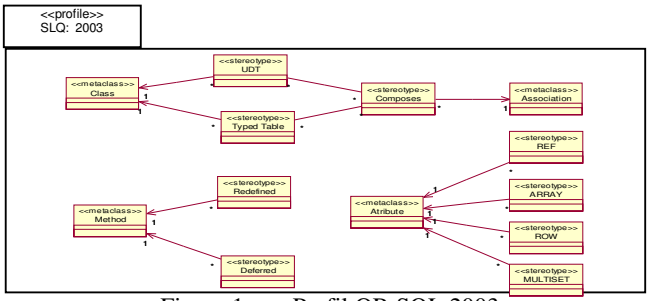


Figura 1 Perfil OR SQL 2003

4 EVALUACIÓN DE HERRAMIENTAS

Como se ha mencionado anteriormente, en la actualidad existe una amplia gama de herramientas que permiten dar soporte a MDA. Se han seleccionado alguna de ellas para realizar una primera evaluación de las mismas.

Para realizar la evaluación se han llevado a cabo las siguientes tareas:

1. Instalación de las distintas herramientas.
2. Instalación de herramientas y componentes adicionales necesarios para su ejecución
3. Implementación del caso de estudio seleccionado:
   a. Definición del perfil UML para BD OR en la herramienta.
   b. Validación del mismo usando una base de datos del OR (un estudio de arquitectura). En cada una de las herramientas se ha comprobado además el código generado en la definición de las clases, atributos y relaciones**.**

A continuación, se explica brevemente el comportamiento de las herramientas seleccionadas en el desarrollo del caso de estudio propuesto. En la Tabla 1 de la sección 5, se resumirán las características analizadas para cada una de ellas.

4.1 Eclipse

Eclipse [5] nace como un entorno de desarrollo para JAVA y se ha convertido en una plataforma que permite el desarrollo y la integración de herramientas de desarrollo. Si bien no puede ser considerada como una herramienta MDA, es una herramienta con la que se pueden generar herramientas MDA. Fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas *VisualAge*. En el año 2001 se forma un consorcio para el desarrollo futuro de Eclipse como código abierto. Finalmente, en el año 2003 este consorcio se independiza de IBM convirtiéndose en la fundación Eclipse.

Se ha incluido Eclipse en este estudio, porque puede ser utilizado, no sólo en el desarrollo de productos software, sino también en el desarrollo de herramientas que permitan construir productos software [5]. Se puede decir también que Eclipse es un marco de trabajo para el modelado y la integración de datos permitiendo almacenar meta-modelos y meta-datos.

La principal ventaja de Eclipse radica en que su arquitectura está diseñada de forma que la mayoría de la funcionalidad proporcionada está localizada en *plug-ins* o en un conjunto de *plug-ins* relacionados. Esto hace a Eclipse una herramienta extensible. Del conjunto de *plug-ins* de Eclipse el presente artículo se centra en *Eclipse Modeling Framework* (EMF) [14]. EMF es una herramienta para la generación de código a partir de modelos definidos por el usuario. A través de ella se pueden

generar *plug-ins* que proporcionen editores de modelos. Caben destacar dos *plug-ins* en concreto, el *plug-in* UML2 permite definir estereotipos de UML y el *plug-in Graphical Modeling Framwork* (GMF) basado en EMF, que permite generar editores gráficos de modelos.

Aplicación del Caso de Estudio

Para el desarrollo del caso de estudio con Eclipse se ha realizado lo siguiente:

- Se ha utilizado la versión 3.2.2 de Eclipse con el *plug-in* de EMF versión 2.3.0.
- Para definir el meta-modelo OR en la herramienta ha utilizado el *plug-in* UML2, para poder representar el perfil, ya que el mismo se define utilizando estereotipos. Una vez definido el perfil con UML2 se ha convertido a un modelo *Ecore* para así poder generar el editor de meta-modelos con EMF. Como EMF no admite la representación de estereotipos en la conversión de UML2 a *Ecore* éstos no han podido ser contemplados. También se ha utilizado el *plug-in* de GMF para generar de forma gráfica editores de modelos.
- Se ha generado el editor de modelos sin los estereotipos. Para corroborar su funcionamiento se ha creado el diagrama de clases correspondiente con la BD del estudio de arquitectura, obteniéndose las clases, atributos y relaciones sintácticamente acordes al meta-modelo OR definido.

Evaluación de características

Conjuntamente con el desarrollo del caso de estudio se han evaluado cada una de las características seleccionadas en el apartado 2.

- *Niveles que cubre:* Utilizando EMF se pueden definir todos los niveles de abstracción definidos en MIDAS, ya que *Ecore* es una implementación de eMOF (*Essential MOF*), por lo que nos permite definir CIMs, PIMs y PSMs
- *Grado de generación de código:* El lenguaje en el que se genera código por defecto, con Eclipse es JAVA, si bien existen otros *plug-ins* que permiten generar código en otros lenguajes. El código generado puede ser verificado y, en caso necesario, se pueden redefinir los modelos y generar el código de nuevo. No se puede realizar ingeniería inversa.
- *Transformaciones:* Se pueden obtener diferentes PSMs a partir del mismo PIM de forma automática. Estos PSMs se pueden pasar al generador de código y se puede generar el código automáticamente [2]. Para realizar las transformaciones entre modelos se debe utilizar *plug-ins* que implementen dichas transformaciones. Para definirlas se puede utilizar cualquier lenguaje que entienda *Ecore*, por ejemplo ATL [3] y QVT [13].
- *Grado de interacción con el usuario:* el usuario participa activamente en las etapas de definición de los meta-modelos y los meta-datos. En las etapas de definición de modelos y datos disminuye la participación del usuario ya que permite tomar como base los meta-modelos definidos con anterioridad, trasladándose de esta manera las características definidas.
- *Tipo de Transformaciones:* únicamente se pueden realizar transformaciones verticales (de PIM a PSM y de PSM a código). Para realizar las transformaciones horizontales se podrían definir nuevos *plug-ins* que las implementen.
- *Lenguaje de almacenamiento y gestión de modelos:* se almacenan en formato XMI. Esto permite definir los diferentes modelos en cualquier herramienta, por ejemplo *Rational Rose,* almacenarlo con formato XMI y exportarlo a EMF.
- *Plataformas y tecnologías soportadas:* soporta la plataforma de JAVA.
- *Ámbito de aplicación:* es una herramienta orientada al desarrollo de sistemas de propósito general. En particular, como hemos dicho anteriormente, en el presente trabajo interesan las facilidades que brinda Eclipse para el desarrollo de software orientado a servicios.
- *Uso de estándares:* utiliza XMI, UML y MOF.
- *Extensibilidad:* se puede extender fácilmente por medio de *plug-ins*, ya que la ventaja principal de Eclipse reside en que cada *plug-in* puede definir puntos de extensión a los que otros *plug-ins*

se pueden conectar, permitiendo de esta manera incrementar la funcionalidad. Así el usuario podría crear el *plug-in* necesario para cubrir nuevos requerimientos.
- *Usabilidad:* es una herramienta amigable. Permite definir los meta-modelos y meta-datos utilizando otras herramientas, por lo que se debe tener en cuenta que el usuario deberá tener conocimiento de las mismas.
- *Interoperabilidad entre herramientas:* se puede adaptar a otras herramientas, por medio de plug-ins existentes o desarrollar nuevos plug-ins.

4.2 AndroMDA

AndroMDA[9] [10] es una herramienta de generación de código que toma modelos UML en formato XMI como entrada y, genera código como salida, en cualquier lenguaje de programación. Nace en el año 2002 como una iniciativa de Matthias Bohlen [1]. En el año 2003 adquiere el nombre de AndroMDA debido a que toma las bases del paradigma MDA.

AndroMDA está compuesta por *cartridges*. Los *cartridges* son un tipo especial de *plug-ins,* donde se definen los meta-modelos y reglas de transformación para transformar elementos del modelo de acuerdo al meta-modelo. En muchos casos, un *cartridge* puede contener solamente los meta-modelos, ya que las reglas de transformación pueden ser manejadas por muchos *cartridges* y pueden ser contenidas en un *cartridge* común.

Se ha seleccionado AndroMDA para este estudio, porque permite la generación de código en diferentes lenguajes de programación.

Aplicación del Caso de Estudio
Para el desarrollo del caso de estudio con AndroMDA se ha realizado lo siguiente:
- El funcionamiento AndroMDA depende del lenguaje con el que se vaya implementar el SI. Para el presente artículo se ha seleccionado como lenguaje de implementación *Visual Studio 2005*, por ello se ha instalado el *cartridge* Android/VS. Además ha sido necesaria la instalación de herramientas adicionales como *MagicDraw 9.5* para realizar los diferentes diagramas y *Apache's Maven* [22] que permite el acceso a las librerías necesarias de AndroMDA de forma transparente para el usuario.
- Para definir el meta-modelo OR se ha utilizado *Magic Draw,* y posteriormente se ha exportado a .Net. En este punto se presentó el mismo inconveniente que se explico en el caso de Eclipse, AndroMDA toma como meta-modelo el meta-modelo de UML. Como el meta-modelo OR utiliza extensiones UML usando estereotipos, que si bien pueden ser representados con la herramienta *Magic Draw,* cuando se exporta a .Net se eliminan los estereotipos. Esto ocurre porque en el *cartridge* de AndroMDA para .Net no están definidos los estereotipos ni la forma de realizar la transformación de los mismos a .Net. Como se ha dicho anteriormente los meta-modelos y las reglas de transformación se encuentran definidas en los *cartridges*, por lo cual para poder desarrollar modelos de clases acordes al meta-modelo OR se debería definir el *cartridge* correspondiente con el meta-modelo y el lenguaje de desarrollo que se desee utilizar.

Evaluación de características
Conjuntamente con el desarrollo del caso de estudio se han evaluado cada una de las características seleccionadas en el apartado 2:
- *Niveles que cubre:* implementa los niveles PIM y PSM, permitiendo realizar transformaciones desde un PIM a varios PSMs. Las transformaciones entre los diferentes modelos se realizan de forma automática.
- *Grado de generación de código:* utiliza la tecnología de *cartridge*, que le permite obtener modelos en diferentes plataformas. A partir de los PSMs se puede generar código en el lenguaje de

programación que se desee siempre que se tenga el *cartridge* correspondiente. Además se debe tener en cuenta que AndroMDA también permite generar código desde un PIM directamente. No se puede realizar ingeniería inversa.

- *Transformaciones:* una vez que se tienen definidos los *cartridges* correctamente, las transformaciones entre los modelos de diferentes niveles son automáticas. El lenguaje utilizado para definir los cartridges en AndroMDA es JAVA.
- *Grado de interacción con el usuario:* se puede decir que la interacción del usuario es alta, ya que debe participar activamente en todas las etapas del desarrollo.
- *Tipos de transformaciones:* únicamente se pueden realizar transformaciones verticales (de PIM a PSM y de PSM a código). Para realizar las transformaciones horizontales se podría, o bien modificar un *cartridge* existente, o definir nuevos *cartridges* que las implementen.
- *Lenguaje de almacenamiento y gestión de modelos:* los modelos generados son almacenados en XMI, permitiendo realizar el intercambio de modelos entre los diferentes niveles [9].
- *Plataformas y tecnologías soportadas:* utiliza herramientas adicionales como *Apache's Maven,* que simplifica su uso, ya que permite a los *cartridges* de AndroMDA para los diferentes lenguajes. Además se pueden utilizar herramientas CASE que permitan generar los diagramas en formato XMI. Utiliza *Nhibernate* para asegurar la persistencia de los objetos. También posee *cartridges* que permiten generar código en plataformas como Java, .Net o PHP.
- *Ámbito de aplicación:* está pensada para el desarrollo orientado a servicios y SI Web.
- *Uso de estándares:* es *MOF-complaint*, permitiendo así soportar estándares como UML y XMI.
- *Extensibilidad:* se extiende creando nuevos *cartridges que* soporten nuevos requisitos.
- *Usabilidad:* en cuanto a la facilidad de uso, se debe considerar que AndroMDA se implementa a través de *cartridges* que se pueden integrar con otras herramientas de desarrollo, por ejemplo Visual .Net o JAVA, por ello es necesario que el usuario posea los conocimientos necesarios en el lenguaje seleccionado.
- *Interoperabilidad entre herramientas:* acepta como entrada los PIMs generados con otras herramientas, como por ejemplo ArgoUML. Además, como se ha dicho anteriormente, tiene *cartridges* definidos para la mayoría de los lenguajes de programación.

4.3 ArgoUML

ArgoUML [18] es una herramienta para el desarrollo de SI. Nace a finales de la década de los 90 como una iniciativa del grupo liderado por Nora Koch. Fue concebido como una herramienta CASE para realizar el análisis y el diseño en el desarrollo de sistemas orientados a objeto y evolucionó hacia los SI Web.

Se ha seleccionado ArgoUML para este estudio, ya que es una herramienta que ha nacido como propuesta de la comunidad de desarrolladores *Open Source*. Está desarrollado utilizando el lenguaje JAVA, permitiendo así que funcione en cualquier plataforma que tenga instalada JAVA2.

Aplicación del Caso de Estudio

Para el desarrollo del caso de estudio con ArgoUML se ha realizado lo siguiente:

- Se ha utilizado la versión 0.24 de ArgoUML. No se requiere ninguna herramienta adicional para el desarrollo del caso de estudio, los diagramas de clases se generan utilizando ArgoUML.
- Se ha comenzado definiendo el perfil OR utilizando la herramienta. En este caso también se ha tenido el mismo inconveniente que en las herramientas anteriores: no se han podido definir los estereotipos del perfil. Como ArgoUML es una herramienta *OpenSource* se puede agregar la funcionalidad correspondiente para poder definir estos estereotipos con la herramienta e implementar las transformaciones correspondientes.

Evaluación de características
Conjuntamente con el desarrollo del caso de estudio se han evaluado cada una de las características seleccionadas en el apartado 2:

- *Niveles que cubre:* cubre todas las etapas del ciclo de vida de un sistema, por lo que se pueden definir modelos para el nivel CIM, PIM y PSM y posteriormente su paso a código.
- *Grado de generación de código:* genera código en el lenguaje JAVA por defecto, aunque existen módulos auxiliares que permiten generar código en lenguajes como C#, C++, PHP4 y PHP5. No se puede realizar ingeniería inversa.
- *Transformaciones:* brinda soporte para realizar transformaciones de PIM a PSM y de PSM a código. Las transformaciones entre los diferentes modelos se realizan de forma automática de acuerdo a lo definido por el usuario. Como se ha dicho anteriormente, ArgoUML está definido en JAVA, por lo que las transformaciones deben definirse utilizando ese lenguaje.
- *Grado de interacción con el usuario* el usuario debe participar activamente en la definición de los diferentes modelos.
- *Tipo de transformaciones:* únicamente se pueden realizar transformaciones verticales (de PIM a PSM y PSM a código). Para realizar transformaciones horizontales se deberían definir las transformaciones, agregándolas como funcionalidad a la herramienta.
- *Lenguaje de almacenamiento y gestión de modelos:* la información de los diferentes modelos se almacena en formato XMI y los gráficos en PGML (*Precision Graphics Markup Language*) basado en XML.
- *Plataformas y tecnologías soportadas:* para su funcionamiento no necesita ninguna herramienta adicional, utiliza la plataforma de JAVA.
- *Ámbito de aplicación:* ArgoUML está pensada para el desarrollo de SI, concretamente en los SI Web. Con la evolución de los SI Web se adaptó al desarrollo de software orientado a servicios.
- *Uso de estándares:* el repositorio de la herramienta está implementado con JMI (*Interface Metadata* de JAVA) y basado en MOF, tomando la especificación de UML 1.4. También tiene soporte para OCL y XMI.
- *Extensibilidad:* Es un proyecto *Open Source,* al que se le puede seguir agregando funcionalidad.
- *Usabilidad:* Tiene una interfaz gráfica muy fácil de entender y manejar, es muy intuitiva. A través de la herramienta se pueden realizar todas las etapas del desarrollo.
- *Interoperabilidad entre herramientas*: los PIMs realizados con ArgoUML pueden ser utilizados como entrada para AndroMDA como se ha indicado en el apartado anterior.

5 DISCUSIÓN

Tras haber desarrollado el caso de estudio con las diferentes herramientas y haber evaluado las características definidas se ha realizado una valoración individual para analizar en que medida satisfacen las características definidas. Cada una de estas características es valorada con 0 (no cumple), 3 (cumple medianamente), 4 (cumple satisfactoriamente) y 5 (cumple completamente).

En la tabla 1 se resumen las características evaluadas en cada una de las herramientas.

	Eclipse EMF	*AndroMDA*	*ArgoUML*
Características Funcionales			
Niveles que cubre	CIM – PIM – PSM	PIM – PSM	CIM - PIM – PSM
Grado de Generación de código	JAVA principalmente	Cualquier lenguaje. Se debe tener el *cartridge* adecuado	JAVA + AndroMDA
Transformaciones	Completamente	Completamente	Completamente
Grado de Interacción con el Usuario	Alto	Alto	Alto

	Eclipse EMF	*AndroMDA*	*ArgoUML*
Tipos de Transformaciones	Verticales. Se pueden implementar las horizontales	Verticales. Se pueden implementar las horizontales	Verticales. Se pueden implementar las horizontales
Características Técnicas			
Lenguaje de almacenamiento y gestión de modelos	XMI	XMI	XMI
Plataformas y Tecnologías soportadas	Ecore – GenModel – JAVA - H. Case para generar diagramas UML	H. Case generen diagramas UML con formato XMI – Nhibernate – Maven – JBoss	JAVA
Ámbito de Aplicación	Desarrollos orientados a servicios	SIW - Desarrollos orientados a servicios	SIW - Desarrollos orientados a servicios
Características de Calidad			
Uso de estándares	XMI – UML – MOF	XMI – UML – MOF	UML – XMI – SVG – OCL
Extensibilidad	Por medio de pluggins	Por medio de *cartridge*	Integración con herramientas
Usabilidad	Fácil de usar	Difícil de usar	Fácil de usar
Interoperabilidad entre herramientas	Integración con otras herramientas	Integración con otras herramientas	Integración con otras herramientas

Tabla 1: Resultado del análisis comparativo de herramientas MDA

En la siguiente tabla se presenta un resumen del resultado de la valoración individual realizada con cada herramienta:

	Eclipse EMF	*AndroMDA*	*ArgoUML*
Características Funcionales			
Niveles que cubre	5	3	5
Grado de Generación de código	4	4	4
Transformaciones	4	4	3
Grado de Interacción con el Usuario	4	3	3
Tipos de Transformaciones	3	3	3
Características Técnicas			
Lenguaje de almacenamiento y gestión de modelos	5	5	5
Plataformas y Tecnologías soportadas	4	5	3
Ámbito de Aplicación	5	5	5
Características de Calidad			
Uso de estándares	5	5	5
Extensibilidad	4	3	3
Usabilidad	5	3	4
Interoperabilidad entre herramientas	4	4	4
Total	**52**	**47**	**47**

Tabla 2: Valoración individual de las herramientas MDA

Después de analizar la valoración individual de las herramientas, se ha llegado a las siguientes conclusiones:

- Como se puede ver en la Tabla 2, Eclipse es la herramienta que cumple en mejor medida la mayoría de las características seleccionadas.
- Al utilizar Eclipse con el *plug-in* EMF se pueden generar editores de meta-modelos de forma fácil si éstos se basan en UML. Sin embargo, no soporta la definición de estereotipos para definir extensiones UML. Teniendo en cuenta que la metodología MIDAS propone el uso de perfiles UML para la mayoría de sus meta-modelos, es necesario utilizar el *plug-in* UML2 para poder definir los meta-modelos de MIDAS. Además, con el uso del *plug-in* GMF se pueden definir los meta-modelos y los modelos correspondientes facilitando así el modelado al usuario. Se debe tener en cuenta que para definir las transformaciones para los diferentes meta-modelos de MIDAS será necesario realizar *plug-ins* que las implementen utilizando para ello el lenguaje que el usuario desee como se ha explicado anteriomente.
- AndroMDA tiene definidos los *cartridges* para la mayoría de los lenguajes de desarrollo, por lo que la implementación en distintas plataformas es muy fácil de realizar. Los *cartridges* se basan

en el meta-modelo de UML. Si se desea realizar modelos basados en perfiles de UML que utilicen estereotipos, se deben modificar los *cartridges* o crear nuevos. Hay que tener en cuenta que se deberían crear *cartridges* para cada uno de los lenguajes de desarrollo con los que se desee trabajar y para cada meta-modelo que se desee introducir. La modificación o creación de nuevos *cartridges* no es trivial, se debe tener conocimiento del lenguaje para el cual se está desarrollando el *cartridge* y de la manera de definir las transformaciones en el mismo.

- ArgoUML se basa en el meta-modelo de UML. Por tanto, al igual que ocurre con AndroMDA, si los modelos que se desean generar son acordes a UML, éstos son fáciles de implementar. Pero si se desea realizar modelos basados en perfiles UML, se debe codificar la funcionalidad necesaria que permita recoger los estereotipos. Si bien ArgoUML únicamente genera código JAVA, los modelos generados con esta herramienta pueden servir de entrada a otras herramientas, como por ejemplo, AndroMDA.

De las tres opciones evaluadas hasta el momento se considera que Eclipse es la que mejor se adapta a las necesidades de MIDAS como arquitectura de modelos, ya que combinando los *plug-ins* EMF, GMF y UML2 se podrá generar una herramienta que se adapte fácilmente a la arquitectura de modelos de MIDAS. Sin embargo hay que tener en cuenta que se debe realizar *plug*-ins para implementar las transformaciones entre los distintos modelos y el usuario deberá poseer los conocimientos necesarios del lenguaje en el que implemente las transformaciones. También se podrían adaptar AndroMDA y ArgoUML, pero en ambos casos la adaptación de las herramientas implica un esfuerzo de desarrollo adicional, ya que se implementarían no solo los meta-modelos sino también las transformaciones entre los modelos.

6 CONCLUSIONES

En este artículo se han analizado las siguientes tres herramientas: Eclipse, AndroMDA y ArgoUML. El objetivo ha sido determinar si alguna de estas puede ser adaptada para dar soporte a la arquitectura de modelos MIDAS o, si por el contrario, es necesario el desarrollo de una nueva herramienta MDA. Para ello, se han seleccionado características funcionales, técnicas y de calidad. Se han analizado todas las características determinando el cumplimiento de las mismas. Además, para evaluar el funcionamiento de las herramientas se ha desarrollado el caso de estudio definido con cada una de ellas.

Este estudio ha permitido concluir, como se ha dicho en el apartado anterior, que de las herramientas evaluadas hasta el momento, Eclipse es la que mejor se adapta a nuestras necesidades. Eclipse complementado con los *plug-ins* de EMF, GMF y UML2 permite de manera sencilla y gráfica generar editores de modelos en base a los meta-modelos definidos por MIDAS.

Actualmente se está trabajando en la evaluación de las herramientas con otros casos de estudio, es decir, con el resto de los meta-modelos propuestos por MIDAS para completar el análisis que se está llevando a cabo.

Esto nos permitirá especificar la arquitectura de la herramienta e implementar un meta-editor de los meta-modelos propuestos por MIDAS.

Agradecimientos

Esté artículo está enmarcado en el proyecto GOLD (TIN-2005-0010) financiado por el Ministerio de Ciencia y Tecnología de España y en el proyecto FOMDAS (URJC-CM-2006-CET-0387) financiado por la Comunidad de Madrid y la Universidad Rey Juan Carlos.

Referencias

[1] *An Interview with Matthias Bohlen.* http://www.codegeneration.net. Marzo 2004.
[2] Backansky, V., *Mastering Eclipse modelling Framework, Tutorial.* EclipseCON. Febrero 2005.
[3] Bézivin, J., Valduriez. P., Jouault, F. *The ATL home page.* http://www.sciences.univnantes.fr/lina/atl.
[4] Fuentes, L., Vallecillo, A. *Una introducción a los perfiles UML.* Novatica Nro.168. Marzo2004.
[5] Gallardo, D. Burnette, E. y McGorven, R. *Eclipse in action, a guide for Java Developers.* Manning Publications. Septiembre 2002.
[6] García Molina, J. Rodríguez, J. Menárguez, M. Ortín, M.J. y Sánchez, J. *Un estudio comparativo de dos herramientas MDA: OptimalJ y ArcStyler.* I Taller sobre desarrollo dirigido por modelos, MDA y aplicaciones (DSDM'2004), noviembre 2004. España.
[7] *ISO / IEC 9075 Standard, Information Technology – Database Languages – SQL:2003.* International Organization for Standardization, 2003.
[8] King´s College London. *An Evaluation of Compuware OptimalJ Professional Edition as an MDA Tool.* University of York, 2003.
[9] Kontio, M. *Architectural manifiesto: MDA in Action.* http://www128.ibm.com-/developerworks/library/wiarch 19/index.html. Octubre 2005.
[10] Kozikowski, J. *A bird's eye view of AndroMDA*. http://galaxy.andromda.org/docs-3.1/contrib/birds-eye-view.html. Noviembre 2005.
[11] Marcos, E. Vela, B., Cáceres, P. y Cavero, J.M. *MIDAS/DB: a Methodological Framework for Web Database Design.* DASWIS 2001. LNCS 2465, Springer-Verlag, septiembre, 2002.
[12] Marcos, E., Vela, B. y Cavero J.M. *Methodological Approach for Object-Relational Database Design using UML. Journal on Software and Systems Modeling (SoSyM).* Springer-Verlag. Ed.: R. France and B. Rumpe. Vol. SoSyM 2, pp.59-72, 2003.
[13] *MOF QVT Standard Specification.* http://www.omg.org/docs/ptc/05-11-01.pdf
[14] Moore, B. Dean, D. Gerber, A. Wagenknecht, G. y Vanderheyden, P. *Eclipse Development using the Graphical Editing Framework and the Eclipse Modeling Framework.* http://ibm.com/redbooks. Febrero 2004.
[15] Naveed Ahsan Tariq and Naeem Akhter. *Comparison of Model Driven Architecture (MDA) based tools (A Thesis document).* Junio, 2005
[16] *OMG. MDA Guide Version 1.0. Document number omg/2003-05-01*. Ed. Millar, J y Mukerji, J. http://www.omg.com/mda.
[17] *OMG, UML Superstructure 2.0.* OMG Adopted Specification ptc/05-07-04. http://www.uml.org/, 2005.
[18] Ramírez, A. Vanpeperstraete, P. Rueckert, A. Odutola, K. Bennett, J. Tolke, L. y van der Wulp, M. *ArgoUML User Manual: A tutorial and reference description.* http://www.opencontent.org/openpub/ 2006
[19] Quintero, J.. Anaya, R. *Marco de Referencia para la Evaluación de Herramientas Basadas en MDA.* IDEAS 07, Mayo 2007, Venezuela.
[20] Vara, J. M., Vela, B., Cavero, J. M., y Marcos, E. *Model Transformation for Obtect-Relational Database development.* ACM Symposium on Applied Computing 2007. Korea, Marzo, 2007.
[21] Vela, B., Cáceres, P., de Castro, V., Marcos, E. *Midas: una aproximación dirigida por modelos para el desarrollo ágil de sistemas de información web.* Ingeniería de la web y patrones de diseño. Capítulo 4. Pearson – Prentice Hall. 2005
[22] *What is Maven?.* http://maven.apache.org/what-is-maven.html.

AUTÓMATAS CELULARES SOBRE GRAFOS DE NODOS IRREGULARES: APLICACIÓN A LA SIMULACIÓN DE ESCURRIMIENTOS SUPERFICIALES EN ZONAS DE LLANURA.

Pablo R. Rinaldi*, Diego D. Dalponte*, Marcelo J. Vénere†, Alejandro Clausse†

**PLADEMA-CICPBA, Universidad Nacional del Centro*
7000 Tandil, Argentina, {ddalpont, prinaldi}@exa.unicen.edu.ar
†CNEA-CONICET, Universidad Nacional del Centro
7000 Tandil, Argentina, {venerem, clausse}@exa.unicen.edu.ar

Resumen

El modelo AQUA-GRAPH [4] de escurrimiento superficial específico para zonas de llanuras se basa en la tecnología de autómatas celulares (AC), de amplia utilización en la simulación de procesos naturales. Este modelo se implementa sobre una estructura de datos grafo.
Este diseño da una mayor flexibilidad al modelo de escurrimiento y al simulador permitiendo realizar diferentes tipos de discretizaciones con el objeto de mejorar la performance y dar diferentes niveles de detalle a la simulación.
En el presente trabajo se perfeccionó el autómata AQUA-GRAPH para poder trabajar con células de diferente tamaño teniendo en cuenta el área y el volumen de agua acumulado en cada celda. Sobre este esquema se desarrolló un algoritmo para discretizar el terreno agrupando conjuntos de celdas con similares características en celdas de mayor área.
Se utilizaron diferentes criterios para fusionar celdas como, diferencia de altura, gradiente acumulado y curvatura acumulada. Se evaluaron estas modificaciones sobre escenarios reales y analizando la sensibilidad a los diferentes parámetros. Los resultados permitieron mejorar notablemente la performance de las simulaciones al reducir la cantidad de elementos a tratar sin perder precisión o nivel de detalle en las zonas más importantes.

Palabras claves: Modelo hidrológico distribuido, Autómatas Celulares, Grafos, Escurrimiento superficial en llanuras.

1 INTRODUCCIÓN

En los cincuenta, Ulam y Von Neumann [2] desarrollaron una ingeniosa herramienta matemática denominada autómata celular (AC) que permite simular ciertos fenómenos complejos con la integración de células finitas, que interactúan de acuerdo a reglas simples basadas en consideraciones heurísticas. Las reglas de interacción que generalmente se aplican a los vecinos inmediatos, pueden o no guardar una semejanza con las leyes físicas que gobiernen el fenómeno. Sin embargo, para fluidos, se encontró que los promedios estadísticos tendían a la solución de las ecuaciones diferenciales parciales que se saben gobiernan la situación, típicamente la ecuación de Navier-Stokes [9].

Desde entonces, la tecnología de AC ha sido utilizada con éxito para la simulación del movimiento de fluidos sobre una superficie como el modelo de simulación de escurrimiento de lava volcánica SCIARA [3] o el modelo de escurrimiento por inventario hidrológico AQUA [5, 6, 8]. La mayoría de estas aplicaciones representan el espacio como una grilla regular o matriz regular en la que todas las celadas tienen el mismo tamaño. Este tipo de estructuras rígidas imponen muchas restricciones a los modelos, como la imposibilidad de contar con celdas de diferentes formas o tamaños, con el consecuente uso ineficiente de los recursos como memoria y tiempo de CPU.

El modelo numérico AQUA-GRAPH, que ya ha sido evaluado favorablemente sobre escenarios sintéticos y reales, utiliza como representación interna un grafo no dirigido, esta estructura brinda una mayor flexibilidad y permite utilizar técnicas de discretización para mejorar el tiempo de respuesta de las simulaciones o aplicar ecuaciones diferentes a sub-regiones.

En este trabajo se modificó el modelo para poder trabajar con celdas de diferentes tamaños y formas sin aumentar el costo computacional del algoritmo. Con esta idea se pretende discretizar espacialmente la grilla generada desde el Modelo Digital de Elevación (MDE).

Básicamente, para la simplificación de mallas, las técnicas de inserción o borrado de vértices son las más comunes y los sistemas de información geográfica (SIG) constituyen el ejemplo más usual de su aplicación. Una de las técnicas más utilizadas es el empleo de clusters (o agrupaciones) de vértices según la proximidad geométrica [7]. El problema de utilizar directamente alguna de estas técnicas es que no fueron diseñadas específicamente para el modelo de simulación de escurrimiento sino para visualización de superficies como los MDE.

Para el autómata AQUA-GRAPH es muy importante con que criterio definir la proximidad geométrica de los nodos que se van a agrupar, ya que se deben preservar las pequeñas depresiones que actúan como reservorios y que influyen en gran manera en el escurrimiento superficial en zonas de llanura. En este trabajo, se analizan varios criterios para definir la proximidad geométrica y realizar agrupaciones de nodos afectando lo mínimo posible la respuesta del simulador. Estos criterios fueron implementados y evaluados con escenarios reales.

2 EL AUTÓMATA AQUA-GRAPH

La representación interna del AQUA-GRAPH se construye a partir de la grilla del MDE, que consiste en un campo escalar $h(x, y)$, asociado a una grilla bidimensional. El campo $h(x, y)$, representa la coordenada vertical promedio de cada celda espacial. Siguiendo el paradigma de AC, el estado superficial de cada celda está determinado por un escalar $w(x, y)$, que representa en este caso el nivel de agua en la celda (x, y).

Con cada celda de la grilla se construye un nodo del grafo donde se almacenan los niveles de

cota del terreno, niveles de agua y de capacidad de infiltración. Posteriormente se agregan arcos bi-direccionales entre nodos vecinos en los sentidos vertical y horizontal (Figura 1), a los cuales se asocia un costo correspondiente a la resistencia al escurrimiento. Esta resistencia al escurrimiento se calcula como la media de las resistencias superficiales de las celdas que une (vegetación, rugosidad, etc.).

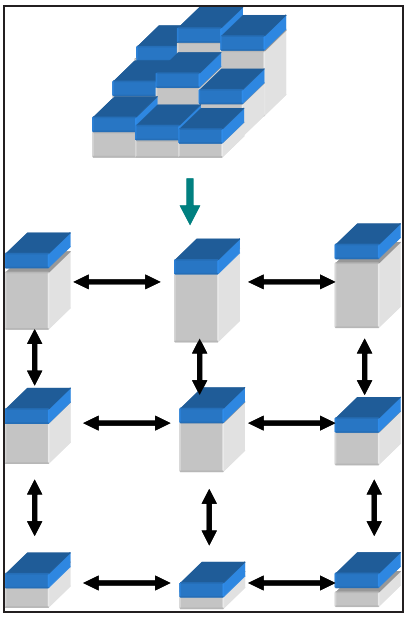


Figura 1: Esquema de construcción del grafo a partir de la grilla.

Para cada nodo del grafo se debe establecer si éste pertenece al terreno o a los cursos de agua. Por último, se recorren todas las celdas del río realizando conexiones diagonales entre celdas para que no se produzcan cortes en el cauce. (Ver Figura 2).

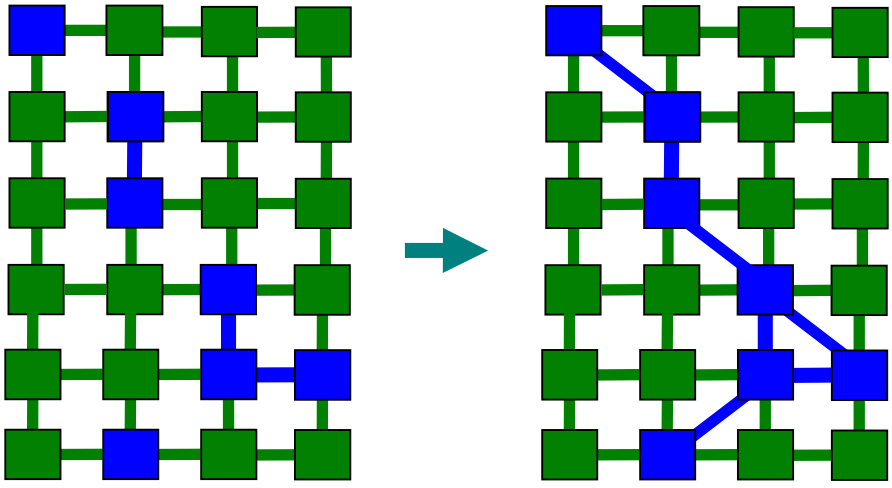


Figura 2: Identificación de nodos y arcos del río en el grafo y reconstrucción de los cauces.

Una vez construido el grafo, cada celda se comporta como un recipiente cerrado que se comunica con otros cuatro, abriendo y cerrando las conexiones dejando que el agua fluya

debido a la diferencias de altura.

2.1 Modelo de escurrimiento superficial

El algoritmo calcula la transferencia de agua entre celdas unidas en base a la ecuación hidráulica estacionaria discretizada en el tiempo. El volumen de agua transferido en un paso depende de las diferencia de alturas totales (terreno más agua) y del costo del arco.

Los pasos del algoritmo de escurrimiento son los siguientes:

1. Para cada celda se crea una lista con las vecinas capaces de recibir agua, o sea, con la altura inferior a la del nodo que se está analizando, llamado “entorno” (Figura 3).
2. Para cada celda y su entorno se calcula cual sería la altura de agua si todo el líquido escurriera a su mínima posición, llamada altura de drenado.
3. Para cada celda se determina la cantidad de agua posible de transferir como la diferencia entre la altura del actual del nodo y la altura de drenado (si este valor es mayor que el total del agua de la celda obviamente el valor se fija en éste último).
4. Se calcula el máximo de agua que le corresponde a cada nodo vecino repartiendo el total transferible (calculado en 3) pesando con la raíz cuadrada de la diferencia de alturas con la celda madre.
5. Finalmente a cada celda se le transfiere un porcentaje del máximo transferible dado por el costo del arco correspondiente (coeficiente de relajación α).

Debe mencionarse que la actualización del nivel de agua se realiza simultáneamente en paralelo en todas las celdas.

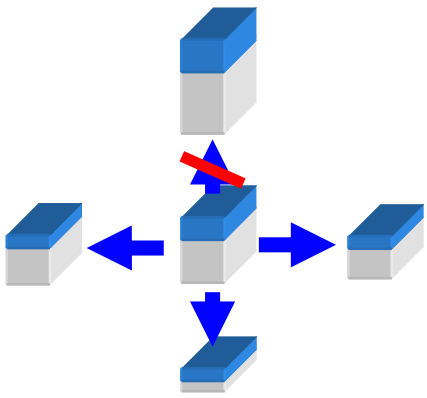


Figura 3: Análisis de una celda (centro) y las vecinas con posibilidad de recibir agua.

Este algoritmo fue modificado para trabajar con celdas de diferentes tamaños. Los cálculos se realizan con volumen de agua (altura de agua multiplicada por el tamaño de la celda) en lugar de altura.

2.2 Fuentes de agua

Los procesos de precipitación e infiltración se modelan como fuentes/sumideros. La precipitación simplemente se agrega a cada celda de acuerdo con un esquema temporal predeterminado, que puede representar datos observados o escenarios hipotéticos.
El proceso de infiltración es más complicado que el de precipitación, cada tasa depende del

estado de saturación del suelo, el cual cambia cuando al infiltrar agua. Para simular este efecto, es necesario registrar el agua infiltrada en cada celda, *I(x, y)*. El volumen de infiltración para el paso de tiempo *n* se calcula como:

$$I_n(x,y) = \begin{cases} 0 \quad si \quad w(x,y) = 0 \\ w(x,y) \quad si \quad 0 < w(x,y) < \beta I_{n-1}(x,y) + I_o(x,y) \\ \beta I_{n-1}(x,y) + I_o(x,y) \quad si \quad w(x,y) \geq \beta I_{n-1}(x,y) + I_o(x,y) \end{cases} \tag{1}$$

donde *Io(x, y)*, es la infiltración base, y β es un coeficiente que representa la característica de saturación del suelo ($0<\beta<1$). Valores pequeños de β significan que el suelo se saturará más rápido. La infiltración inicial *Ii*, viene dada por las condiciones iniciales del suelo y debería ser ingresada por el analista.
Los procesos de infiltración y precipitación no necesitan ser modificados para trabajar con celdas de diferente tamaño ya que influyen solamente en la altura de agua independientemente del área de la celda.

2.3 Modelado de ríos y arroyos

Los ríos y arroyos de agua se simulan en modelo AQUA-GRAPH reduciendo la resistencia al flujo a lo largo de los cursos de agua correspondientes. Por consiguiente, el valor de α para cada arco del grafo que pertenece al río se calcula como una función de la altura de agua promedio de los nodos que comunica, lo que representa la influencia del perfil del lecho del río sobre el flujo.
Como existen numerosos factores que afectan esta relación (perfil del lecho, características del suelo, vegetación acuática, etc.) un modelo integral requeriría la definición de una función para cada celda. Sin embargo, esto es prácticamente imposible cuando se modelan grandes extensiones de terreno. Como alternativa, se pueden definir familias de funciones-α por regiones, cuyos parámetros pueden ser determinados comparando los cálculos numéricos con datos experimentales.
Una familia de funciones que muestra una buena aproximación a las mediciones de caudal de los arroyos del sur de la Provincia de Buenos Aires es la siguiente:

$$\alpha_{rio} = \left(\frac{1}{\left[1 + \left(\frac{H_0}{W} \right)^p \right]^n} \right) \tag{2}$$

donde W es la altura de agua promedio de las celdas conectadas y Ho, p y n son parámetros constantes.

2.4 Condiciones de contorno

Se aplican condiciones de borde abierto al contorno externo de la región simulada. Esto se implementa agregando un borde auxiliar de celdas, con elevación substancialmente inferior a las adyacentes del terreno. El volumen de agua contenido en el borde adicional es eliminado a cada paso del algoritmo para evitar acumulaciones a largo plazo. Para lograr un correcto balance de masas global, es muy importante que el dominio de simulación sea más grande que

la cuenca en estudio, para permitir que los límites de cuenca se determinen automáticamente.

2.5 Escalas temporales diferentes:

La estructura de grafo permite simular de manera separada el escurrimiento en el terreno y en el río utilizando diferentes escalas temporales y estableciendo puntos de sincronización donde los resultados de ambas simulaciones son actualizados entre si. El grafo principal se divide en dos sub-grafos con arcos comunes entre sí y por cada ciclo del grafo del terreno, se realizan varios ciclos de simulación en el grafo del río.
Dado que la cantidad de celdas de río es generalmente menor al 1% del total, al aumentar la escala temporal a las celdas del terreno, se logra una aceleración casi lineal del proceso global de la simulación.
En la práctica, se utiliza la misma relación temporal para el grafo del río, mientras que se reduce la cantidad de pasos en el grafo del terreno en una relación que de ahora en más denominaremos aceleración. Para obtener resultados equivalentes, al aplicar una aceleración al terreno, se debe aumentar en la misma relación el valor de alfa utilizado. Esto pone un límite a la aceleración ya que el valor de alfa debe ser menor o igual a uno.
Para las simulaciones del presente estudio se utilizó siempre un valor de aceleración de 10.

3 DISCRETIZACIÓN DE LA REPRESENTACION EN NODOS IRREGULARES

Una de las características más importantes que diferencia al modelo AQUA-GRAPH de los clásicos autómatas celulares de grilla es que permite representar conjuntos de celdas en un solo nodo. Esto se debe a que la representación mediante grafo permite separar las relaciones de vecindad de las características morfológicas del área representada por un nodo. Esto permite dar diferentes niveles de detalle espacial al MDE.

3.1 Agrupación de nodos:

Se desarrolló un algoritmo que busca similitudes entre nodos vecinos para agruparlos siguiendo un criterio dado. El algoritmo recorre toda la lista de nodos del grafo buscando unir a cada nodo con alguno de sus vecinos. Si encuentra un par que cumple con el criterio los une creando un nuevo nodo con las características de los agrupados. Este nuevo nodo conserva los vecinos de los dos que agrupa, y si tenían algún vecino común el nuevo nodo lo tendrá pero conectado con la suma de las capacidades de las conexiones. El arco que unía a los nodos originales desaparece. Por último se eliminan los nodos originales y el nuevo se agrega al final de la lista, pudiendo volver a ser agrupado con otros vecinos cuando el algoritmo llegue a ese punto de la lista. El algoritmo termina cuando se recorrió toda la lista de nodos.

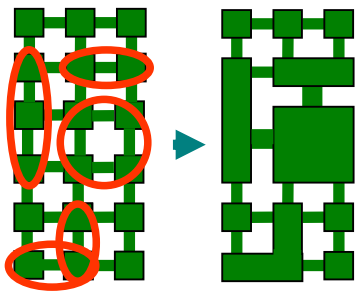


Figura 5: Esquema de agrupación de nodos, las conexiones más anchas tienen el doble de capacidad.

Las características de un nodo nuevo que agrupa a dos existentes se calculan de la siguiente

manera:

1. El Área es igual a la suma de las áreas de los nodos agrupados.
2. La cota de terreno es igual a la suma de alturas de terreno de cada nodo multiplicada por el área, dividido el área del nuevo nodo.
3. El volumen de agua es igual a la suma de los volúmenes. Para el caso de que en el estado inicial de la simulación el nivel de agua de alguno de los nodos no sea cero.

Este tipo de unión se realiza solamente con los nodos del terreno que son los que requieren menos nivel de detalle. Los nodos de río representan menos del 1% del grafo y tienen conexiones que recalculan sus costos dinámicamente.

3.2 Criterios de Agrupación de nodos:

Se implementaron varios criterios de agrupación de nodos y se evaluaron los resultados.

Diferencia de altura:
Como el modelo de escurrimiento trabaja en base a la cota de altura de las celdas, nodos vecinos que tienen la misma altura trabajan como uno solo ya que no hay transferencia entre ellos si la altura de agua es la misma. Entonces, siguiendo este criterio, si la diferencia de altura entre nodos vecinos es inferior a un coeficiente dado, los nodos se unen.

Gradiente acumulado:
El gradiente es un vector que caracteriza el comportamiento del campo escalar de alturas *h(x, y)* en el entorno de la celda ubicada en la fila y columna *x, y* que se expresa en la ecuación (3) cuyas componentes son las derivadas parciales de la función en ese punto. La ecuación (4) expresa un esquema centrado de diferencias finitas donde Δx y Δy representan la distancia entre los pares de celdas vecinas a la celda.

$$grad(h) = \left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y} \right) = \left(G_x, G_y \right) \tag{3}$$

$$G x_{i,j} = \frac{\partial h_{i,j}}{\partial x} = \frac{h_{i+1,j} - h_{i-1,j}}{2\Delta x} \tag{4}$$

$$G_{y_{i,j}} = \frac{\partial h_{i,j}}{\partial y} = \frac{h_{i,j+1} - h_{i,j-1}}{2\Delta y} \tag{5}$$

Para cada nodo del grafo se calcula el módulo del vector gradiente, esta magnitud nos mostrará que tan empinada se encuentra la pendiente. Para este criterio, si la celda se encuentra en una zona de poco pendiente se puede agrupar con celdas vecinas. Entonces, cada vez que dos nodos se agrupan, se acumulan los módulos de sus gradientes en el nuevo nodo. El gradiente acumulado en cada nodo no debe superar un valor dado como parámetro.

Curvatura acumulada:
Frecuentemente, se recurre al concepto de curvatura para caracterizar la geometría local de mallas de superficie [1]. En geometría diferencial, la curvatura representa la apariencia local de una dada superficie. En este trabajo se aplican criterios isotrópicos que evalúan una forma de la curvatura media (la curvatura local). Las componentes del tensor curvatura se calculan

también en forma centrada como:

$$\kappa_{i,j} = \frac{\begin{bmatrix} \left(Gx_{i+1,j} - Gx_{i-1,j}\right) & \left(Gy_{i+1,j} - Gy_{i-1,j}\right) \\ \left(Gx_{i,j+1} - Gx_{i,j-1}\right) & \left(Gy_{i,j+1} - Gy_{i,j-1}\right) \end{bmatrix}}{2\Delta} \quad (6)$$

donde se asume que $\Delta x = \Delta y = \Delta$.

Dado que en nuestra aplicación interesa conocer solo la magnitud de la curvatura y no sus componentes individuales, se utilizó el siguiente indicador:

$$\kappa^2_{i,j} = \left(Gx_{i+1,j} - Gx_{i-1,j}\right)^2 + \left(Gy_{i+1,j} - Gy_{i-1,j}\right)^2 + \left(Gx_{i,j+1} - Gx_{i,j-1}\right)^2 + \left(Gy_{i,j+1} - Gy_{i,j-1}\right)^2 \quad (7)$$

Una vez calculada la curvatura *K* para cada nodo, se procede de manera similar al modo del gradiente acumulado para agrupar nodos.

Criterio combinado:

Como el objetivo de este estudio es conseguir representar más eficientemente MDEs de zonas de llanura para simular procesos hidrológicos con el modelo AQUA-GRAPH, es muy importante que en el proceso de discretización no se pierdan las pequeñas depresiones del terreno que actúan como reservorios de agua y que influyen de gran manera en el escurrimiento superficial. Al observar la imagen del MDE luego de una simulación o una imagen satelital notamos que la mayor concentración de depresiones inundadas se encuentra en la zona baja de la cuenca, donde la pendiente es menor. Por lo tanto, las pequeñas depresiones del terreno (valor de curvatura alta) que están ubicadas en la zona de menor pendiente (valor de gradiente bajo cercano a cero) no deberían agruparse. En cambio, celdas vecinas con valor de curvatura baja que están en una zona de pendiente (gradiente alto) pueden agruparse con sus vecinas ya que no forman reservorios.

El criterio combinado se define calculando para cada nodo el coeficiente:

$$C_{x,y} = K_{x,y} \frac{1}{G_{x,y}} \quad (8)$$

donde $K_{x,y}$ es el indicador de curvatura y $G_{x,y}$ es el módulo del gradiente. Cada vez que dos nodos se agrupan, se acumulan sus coeficientes *C*. El *C* acumulado en cada nodo no debe superar un valor dado como parámetro.

4 EVALUACIÓN DE LOS DIFERENTES CRITERIOS

4.1 Escenario real utilizado

Para evaluar los diferentes criterios de agrupación de nodos, el modelo AQUA-GRAPH se aplicó sobre el MDE de una cuenca real con datos reales de un evento de tormenta. No se utilizaron modelos sintéticos como planos inclinados ya que se buscaba discretizar terrenos irregulares con pequeñas depresiones y evaluar su impacto sobre el modelo de escurrimiento.

La cuenca utilizada está ubicada en el centro de la provincia de buenos Aires, Argentina. Entre los 36º 8’ y los 37º 22’ de latitud sur y entre los 58º 49’ y 60º 10’ de longitud Oeste. Su extensión es de aproximadamente 150 Km. en la dirección Sudoeste-Noreste y de 40 Km. de ancho compuesta por una gran planicie con pequeñas sierras en el límite sur, donde nace el río

Azul. La pendiente promedio es del 5% en la zona sur y de menos del 0,2% en la llanura 5.

Como dato *input* de la geometría de la cuenca se utilizó un MDE con mediciones de interferometría radar construidas de imágenes satelitales correspondientes a la misión ERS Tandem en 1997. Los detalles del procesamiento involucrado en la construcción del MDE se describen en [10]. El tamaño de celda del modelo resultante es de 80 m. En la región baja se observan numerosas depresiones, las cuales serán responsables más adelante de los estancamientos de agua.

La región modelada en este estudio es la cuenca del arroyo Santa catalina, que es un tributario del Río Azul. La longitud de la cuenca es de 32 Km., y drena 158 km2 de los sectores superior y medio de la cuenca del río Azul. La pendiente típica en la cuenca alta es de entre 1 y 10 % (afloramientos rocosos), mientras que valores del 0,1 a 1 % caracterizan la zona media.
Para este estudio se utilizó un evento de inundación ocurrido en Agosto de 2002. Se cuenta con los siguientes datos hidrológicos del evento:

- Precipitación acumulada proveniente de pluviómetros distribuidos a lo largo de la cuenca.
- Hydrograma de la cuenca Santa Catalina calculado de la curva HQ utilizando datos de nivel de agua medidos en la salida de la cuenca.
- Nivel freático previo a la lluvia.

Los parámetros de escurrimiento e infiltración utilizados en las simulaciones son los mismos que en [4].

4.2 Comparación.

Se definieron varios indicadores para poder evaluar los diferentes criterios, estos son:

Porcentaje de nodos: Para evaluar el grado de discretización alcanzado, se cuenta la cantidad de nodos con la que se representa la cuenca luego de la discretización. Cuanto menor sea el porcentaje sobre la cantidad de celdas originales mayor será el nivel de discretización alcanzado.
Volumen total eliminado: El volumen total de agua eliminado por el río al finalizar la simulación nos permite medir si se perdieron algunas de las pequeñas depresiones que actuaban como reservorio y por lo tanto el agua que debía ser retenida por la cuenca drenó hacia el río. Este volumen se calcula de manera aproximada en base a las mediciones de caudal que se tomadas una por hora.
Hidrogramas de salida: Se analizaron los hidrogramas de salida en comparación con las mediciones originales y las simulaciones con el MDE original con el objeto de evaluar la precisión.

Para cada criterio, se buscó el mayor porcentaje posible de discretización con la menor diferencia en los hidrogramas de salida y el volumen eliminado. Los resultados se muestran en la Tabla 1.

	Agrupamiento			Volumen Eliminado
Criterio	**Coeficiente de Agrupamiento**	**Cantidad Nodos**	**Porcentaje del total**	**m^3**
Sin Agrupar	-	58.394	100 %	3.089.875
Diferencia de Altura	0,1	48.700	83 %	3.163.868
Gradiente Acumulado	0,01	55.337	95 %	3.234.556
Curvatura Acumulado	0,00015	51.289	87 %	3.195.020
Combinado	0,03	27.265	45 %	3.122.149

Tabla 1: Discretizaciones máximas para los diferentes criterios.

Las figuras 8, 9, 10 y 11 muestran el hietograma de precipitaciones en barras correspondiente a los milímetros de agua promedio por hora. Los puntos sólidos en los gráficos representan el flujo de descarga de la cuenca al río principal.

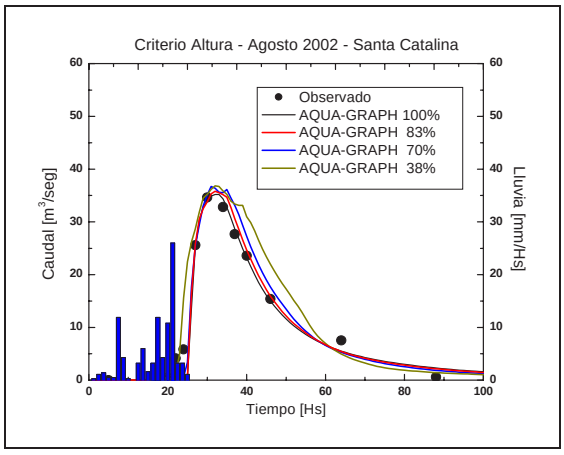


Figura 8

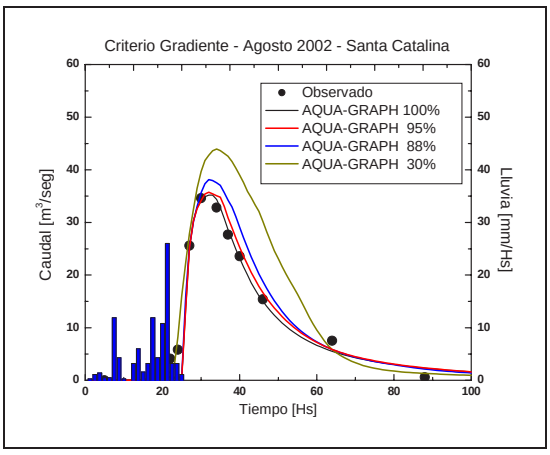


Figura 9

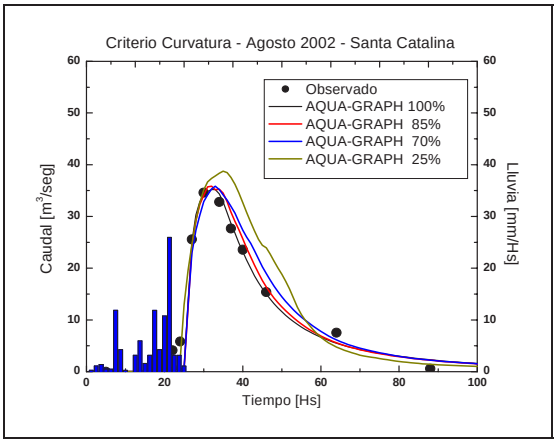


Figura 10

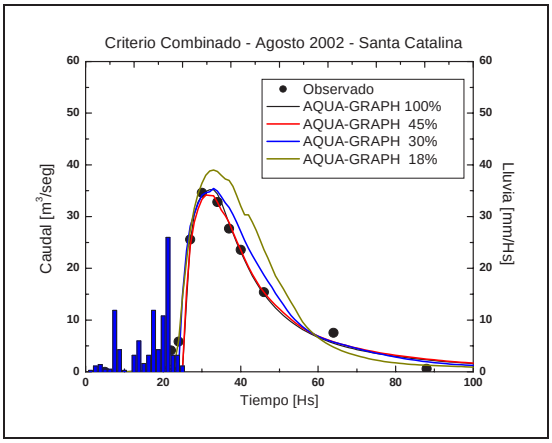


Figura 11

Figuras 8, 9, 10 y 11: Caudal observado (puntos) y calculado (curva) en la salida de la cuenca del Santa Catalina para los criterios de agrupación por diferencia de altura, gradiente acumulado, curvatura acumulada y criterio combinado respectivamente.

Al analizar los resultados vemos que con los primeros tres criterios, la cantidad de nodos totales necesarios para no perder precisión debe ser mayor al 80%. Por debajo de estos niveles, los hidrogramas muestran curvas más altas y anchas lo que se traduce en un mayor volumen de agua eliminado por el río. Esto se debe principalmente a que las pequeñas depresiones desaparecen, sobre todo en las zonas de menor pendiente. En cambio, con el criterio de agrupación combinado se logró una muy buena simulación con el 45% de los nodos y una bastante aproximada con sólo el 30% de los nodos. Si bien estos valores son específicos para esta cuenca, permiten ver la mejora con respecto a los otros criterios implementados.

5 CONCLUSIONES

Se adaptó el modelo de escurrimiento AQUA-GRAPH para trabajar con celdas de diferentes tamaños utilizando el volumen para calcular las transferencias entre celdas y se implementó un algoritmo que agrupa nodos del grafo con características similares.
Se analizaron e implementaron diferentes criterios de unificación como diferencia de altura, gradiente acumulado y curvatura acumulada y se evaluaron los resultados sobre el MDE de una cuenca real con datos de una tormenta real. A partir del comportamiento de los primeros criterios, se definió uno combinando Gradiente y Curvatura que mostró dar muchos mejores resultados. Para la cuenca del ejemplo se logró reproducir de manera muy precisa el hidrograma de salida con menos del 50% de los nodos originales. Y se obtuvo una muy buena respuesta aproximada con el 30% de las celdas.
La discretización espacial implementada, mejora notablemente la performance ya que permite reducir la cantidad de elementos a tratar en cada paso de la simulación sin aumentar el costo computacional del algoritmo de escurrimiento. El modelo AQUA-GRAPH con la modificación para trabajar con celdas de diferentes tamaños permitiría también construir un grafo único con varios MDEs de diferentes resoluciones unidos.

6 AGRADECIMIENTOS

Este proyecto fue financiado por la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CICPBA) y la Comisión Nacional de Energía Atómica (CNEA). Las imágenes satelitales fueron cedidas por la Comisión Nacional de Actividades Espaciales (CONAE). Los autores desean agradecer especialmente el apoyo del Instituto de Hidrología de Llanuras (IHLLA), responsable de los datos mostrados en este artículo.

REFERENCIAS

[1] Cifuentes, M.V., D' Amato, J.P., García Bauza, C., Vénere, M.J., Clausse, A., *Remallado interactivo como método para la navegación eficiente de modelos topográficos*, Mecánica Computacional, Vol. XXIV. ISSN 1666-6070. pp. 249. 2005
[2] Cooper, N.G. (ed.). *Los Alamos Science Special Issue, 1987*. Los Alamos National Laboratory, New México. 1987.
[3] Crisci, G., Rongo, R., Di Gregorio, S., Spataro, W., *The simulation model SCIARA: the 1991 and 2001 lava flows at Mount Etna*. Journal of Volcanology and Geothermal Research 3010 1-15. 2004
[4] Dalponte, D., Rinaldi, P., Vénere, M. Clausse, A. *Algoritmos de grafos y autómatas celulares: Aplicación al la simulación de escurrimientos.* Mecánica Computacional, Vol. XXIV. ISSN 1666-6070. pp. 19. 2005

[5] Dalponte D., Rinaldi P., Cazenave G., Usunoff E., Varni M., Vives L., Vénere M.J., Clausse A. *A validated fast algorithm for simulation of flooding events in plains.* Hydrological Processes. 21: 115-1124. 2007.

[6] Rinaldi P., Dalponte D., Vénere M., Clausse A. *Cellular automata algorithm for simulation of surface flows in large plains.* Simulation Modeling Practice and Theory. 15: 315-327. 2007

[7] Rossignac, J. Borrel, P. *Multiresolution 3D approximation for rendering complex scenes.* Geometric Modeling in Computer Graphics, Falcidieno B. and Kunii T.L. (eds.). Springer Verlag, pp. 455-465.1993

[8] Vénere, M. J., Clausse, A., Dalponte, D., Rinaldi, P., Cazenave, G.., Varni, M., Vives, L., Usunoff, E. *Simulación de Inundaciones en Llanuras. Aplicación a la Cuenca del Arroyo Santa Catalina-Azul*. Mecánica Computacional, Vol. XXIII, pp.1135-1150. 2004.

[9] Wolfram, S. *Cellular automata fluids*. Journal of Statistical Physics 45, 3-4. 1986

[10] Euillades, P., Vénere, M. *Corrección de modelos de elevación en base a un conjunto de puntos seguros*. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería, 19, 33-43. 2003.

Desenvolvimento de Sistema Computacional para auxílio à Reabilitação de Distúrbios Auditivos Leves ou Moderados

Elisangela Maschio de Miranda, Anita Maria da Rocha Fernandes, Leonardo Baptista Lopes, Willian Norberto Westphal, Fernanda Debortoli , Raphael Poolney Teixeira de Jesus
Laboratório de Inteligência Aplicada – LIA (Universidade do Vale do Itajaí – UNIVALI)
Itajaí, Santa Catarina, Brasil

Sheila Andreoli Balen, Simone Mariotto Roggia, Karla Jean Zimmerman, Sinara dos Santos, Mariane Perin da Silva, Ademir Antonio Comerlatto Junior , Alex Mauricio Demarchi Trombelli
Instituto de Fonoaudiologia (Universidade do Vale do Itajaí – UNIVALI)
Itajaí, Santa Catarina, Brasil

Abstract

This paper presents the development of a system to reduce/eliminate language and learning difficulties of low or moderate hearing impaired children. The Software Auxiliar na Reabilitação dos Distúrbios Auditivos - SARDA has five games to develop this hearing skills: to locate the sound, recognition sound, selective and supported attention. The games are organized in stages, phases and levels with differences acoustics related to the sounds time, verbal or not verbal. To the phonoaudiology the software is to develop the hearing skills at hearing impaired children, and to the teacher is to develop this hearing skills in children at school development. Both can use the software and to have informations about the evolution of the child in the treatment.

Keywords: Games, Language e Learning Difficulties, Hearing Impaired.

Resumo

Este artigo apresenta o desenvolvimento de um software com o objetivo de reduzir/eliminar dificuldades de linguagem e aprendizagem que deficientes auditivos e pessoas portadoras de distúrbios do processamento auditivo venham a possuir. O Software Auxiliar na Reabilitação dos Distúrbios Auditivos – SARDA - é constituído por cinco jogos que visam desenvolver, principalmente, as seguintes habilidades auditivas: localização da fonte sonora, reconhecimento e discriminação auditiva, atenção seletiva e sustentada. Os jogos estão organizados em etapas, fases e níveis de dificuldades que têm como foco diferenças acústicas relacionadas ao tempo dos sons não verbais e verbais. Desta forma, a linha mestre entre os jogos é auxiliar a criança a ser mais eficiente e eficaz em processar os estímulos auditivos no tempo. O software tem como objetivo principal para o fonoaudiólogo desenvolver estas habilidades auditivas em crianças com distúrbios auditivos, enquanto que para o professor o foco principal é desenvolver estas habilidades auditivas em crianças em fase de desenvolvimento escolar. Ambos os profissionais podem utilizar o software e obter um parecer a respeito da evolução ou não da criança no tratamento.

Palavras-chave: Jogos, Dificuldades de linguagem e aprendizagem, Distúrbios Auditivos.

1. INTRODUÇÃO

A deficiência auditiva acomete 16,7% da população segundo dados do [3]. De acordo com [4], a perda auditiva é uma desvantagem, pois pode levar ao atraso no desenvolvimento da fala e da linguagem, problemas sociais e emocionais, bem como ao fracasso escolar. Os autores ainda destacam que todos esses problemas podem ser minimizados, pois um tratamento efetivo reduz significativamente a incapacidade da perda auditiva.

Dentre os graus e tipos de deficiência auditiva existe uma preocupação referente às perdas auditivas leves e moderadas e aos distúrbios de processamento auditivo, pois são mais difíceis de serem detectados e pode gerar prejuízos ao desenvolvimento da audição, atenção, memória e linguagem das pessoas que as possuem, sendo que por vezes podem produzir impactos sociais, educacionais e psicológicos importantes.

Os portadores de distúrbios do processamento auditivo apresentam limiares auditivos dentro dos padrões da normalidade, mas têm dificuldade em uma ou mais das habilidades auditivas: atenção seletiva e sustentada, memória de curto prazo e memória operacional, integração e separação binaural das informações. O não diagnóstico e tratamento destes casos pode acarretar várias dificuldades no processo escolar, sendo em grande parte o causador dos insucessos escolares.

De acordo com [10], crianças que possuem alguma deficiência auditiva necessitam de intervenção terapêutica formada por técnicas, estratégias e procedimentos que sejam transformadores, prazerosos e afetivos, o que faz com que a construção de um software interativo possa ser de grande utilidade para esta intervenção.

Dentro deste contexto este artigo apresenta o projeto SARDA, que tem por objetivo o desenvolvimento de um sistema computacional que visa reduzir e/ou eliminar dificuldades de linguagem e de aprendizagem em deficientes auditivos, baseando suas estratégias de estímulo das habilidades auditivas no software Fast ForWord [9; 8]. O SARDA foi desenvolvido utilizando PHP e PostgreSQL, além do Sound Forge para tratamento dos sons.

2. METODOLOGIA

O SARDA foi planejado para ser aplicado em deficientes auditivos com perda auditiva leve e/ou moderada e com distúrbios do processamento auditivo, a fim de minimizar ou sanar alterações de linguagem e de aprendizagem nestes sujeitos. Para atingir o objetivo do sistema foi necessário dividir seu desenvolvimento em três momentos: (i) metodologia de desenvolvimento de software; (ii) metodologia de aplicação do software no contexto terapêutico; e (iii) metodologia de aplicação do software no contexto educacional.

Para o desenvolvimento do sistema foi seguida a seguinte metodologia: (i) levantamento de requisitos do software; (ii) análise; (iii) projeto; (iv) implementação; (v) validação e testes; (vi) implantação.

O levantamento de requisitos foi realizado através de reuniões e entrevistas com profissionais do setor de Fonoaudiologia, e a análise do software Fast ForWord por todos os integrantes do projeto. A partir das informações levantadas estabeleceu-se quatro categorias de usuários: (i) administrador; (ii) fonoaudiólogo; (iii) professor; e (iv) criança. De acordo com a categoria que o usuário for cadastrado ele tem direito a acessar determinada área, e esse acesso é realizado através de *login* e *senha*, conforme Figura 1.

Figura 1: Tela de Entrada

O usuário administrador possui acesso a todos os parâmetros do sistema, mas não pode alterar exames cadastrados pelo fonoaudiólogo e nem alterar resultados dos jogos realizados pelas crianças. Os exames realizados nas crianças podem ser cadastrados e alterados somente pelo fonoaudiólogo, e o professor e administrador possuem somente direito a visualização destas informações. Além disso, o fonoaudiólogo pode cadastrar crianças que sejam suas pacientes, alterar seu próprio cadastro, mas não possui acesso aos dados de pacientes de outros fonoaudiólogos. O usuário fonoaudiólogo tem acesso ao cadastro de dados referentes à criança, tais como: a) aspectos de saúde em geral; desenvolvimento neuropsicomotor; de linguagem, auditivo e escolar, antecedentes familiares e psicopatológicos da criança; b) avaliação fonoaudiológica; c) avaliação audiológica básica, eletrofisiológica e do processamento auditivo. Também pode alterar parâmetros referentes ao jogo, que serão modificados de acordo com a avaliação realizada previamente pelo próprio fonoaudiólogo, podendo ser diferenciada de criança a criança. Já o usuário professor somente acessará o cadastro de seus alunos.

Referente à aplicação do software no contexto terapêutico serão selecionados três grupos de 20 crianças cada, no Setor de Fonoaudiologia da Universidade e na Rede Municipal de Ensino. Os três grupos dividem-se em: A – 20 indivíduos com distúrbios de processamento auditivo; B – 20 indivíduos com perda auditiva leve e/ou moderada; C – 20 indivíduos com limiares auditivos dentro dos padrões da normalidade e sem distúrbio do processamento auditivo. O sistema será aplicado em cada criança após as avaliações de cunho fonoaudiológico, audiológico e referente ao processamento auditivo. A seguir as crianças serão expostas às aplicações do sistema três vezes por semana no período de 50 min, durante doze semanas. Todas as crianças realizarão novas avaliações fonoaudiológicas, audiológicas e do processamento auditivo, bem como avaliação de linguagem e de consciência fonoaudiológica após o total de sessões de aplicação do sistema. Os dados obtidos serão comparados com a finalidade de verificar a efetividade do SARDA.

Quanto ao contexto educacional, será realizado um treinamento sobre o SARDA e o seu uso com um grupo de professores de primeira a terceira série. Após o treinamento será selecionado um dos professores e a sua turma para que o software seja aplicado pelo professor, com monitoramento dos pesquisadores, três vezes por semana durante 50 minutos, ao longo de doze semanas. Anterior a aplicação do software todos os alunos da turma serão avaliados na escola quanto ao desenvolvimento de linguagem, consciência fonológica, audição e processamento auditivo. Após a finalização da estimulação as mesmas serão reavaliadas, no intuito de verificar as possíveis mudanças ocorridas. O intuito da aplicação do SARDA no contexto educacional é visando o desenvolvimento das habilidades auditivas e os testes possuem como foco verificar se o mesmo atinge este objetivo.

Foram pré-definidos cinco jogos para esta primeira versão do sistema, sendo que cada jogo possui as etapas fácil, intermediária e difícil, cada etapa terá fases e cada fase três níveis. Quanto maior a etapa, fase e nível, maior a dificuldade para a criança. O SARDA está sendo desenvolvido utilizando banco de dados PostgreSQL, servidor http Apache SSC, PHP e Flash.

Na sua totalidade, as atividades propostas no SARDA tem por objetivo estimular as habilidades primárias referidas no *Fast ForWord Language*, tendo em vista o desenvolvimento das habilidades necessárias para um adequado aprendizado escolar. Deve-se destacar, no entanto, que cada uma das atividades propostas irá estimular determinados tipos de habilidades auditivas. Para que a criança se sinta motivada durante as atividades a cada etapa cumprida, ela será premiada de acordo com a temática da estratégia e haverá a cada cinco tentativas de sons animações inusitadas que buscarão descontrair a criança e proporcionar que a sua interação com o jogo seja prazerosa. Com o decorrer de finalizações ela irá montando um álbum de fotos dos personagens, dos prêmios e das temáticas que cercam as estratégias.

As atividades do SARDA foram desenvolvidas ambientadas em um parque florestal. Denominou-se o conjunto de cinco jogos como o módulo: Decifrando a audição. Foram pré-definidos cinco jogos, sendo que em cada um a criança interage com animais diferentes. São salvos no banco de dados todos os dados referentes às jogadas realizadas pela criança, para que o fonoaudiólogo possa verificar qual a melhor forma de trabalhar com a mesma.

O primeiro jogo denominado **Localizando com o Mico Mané**, utiliza estímulos não verbais (tom puro na freqüência de 1000 Hz), com diferentes durações e intervalos entre os estímulos. A criança, usando fones de ouvido, tem que localizar em qual lado está sendo apresentado o estímulo. Neste jogo é trabalhada, principalmente, a habilidade auditiva de localização da fonte sonora. Esta habilidade está relacionada com a determinação da direção e distância a que o estímulo sonoro está em relação ao ouvinte. Ocorre, segundo [7], *a partir das diferenças interaurais de tempo, fase e intensidade dos estímulos sonoros que atingem as duas orelhas*, havendo, portando, a necessidade de audição binaural. Segundo [2], a habilidade de localização auditiva contribui com os processos de atenção auditiva e é determinada pelos mecanismos sensoriais e neurais da audição. Para que ela ocorra, requer a habilidade de interpretar diferenças sutis de intensidade, de espectro e de tempo de chegada dos sons às duas orelhas.

No segundo jogo denominado de **Pulando com Dinho Golfinho** a criança tem que discriminar a freqüência do estímulo apresentado. Os tipos de sons utilizados serão estímulos

não-verbais diferindo quanto à freqüência. Os níveis de dificuldade desta estratégia irão variar quanto à diferença entre as freqüências, bem como quanto ao intervalo entre os estímulos, conforme ordem crescente de dificuldade. Nesse jogo é trabalhada, principalmente, a habilidade auditiva de discriminação auditiva, que é definida por [7] como *o processo de diferenciação de sons acusticamente similares, mas com freqüência, duração e/ou intensidade diferentes*. Por ser uma habilidade complexa dependem também de outras funções como a atenção, o conhecimento e a memória, sendo que o processo de discriminação só ocorre *após o padrão ser internamente resintetizado, comparado com o modelo interno e então categorizado* [6].

No terceiro jogo, **Escutando com Tuca Tucano, a** criança ouve um estímulo (som alvo) e em seguida tem que distinguir entre dois outros sons, qual é igual ao som alvo apresentado. Serão utilizados estímulos verbais com estruturas silábicas diferentes – CVC, CCV, VCV. Inicialmente os estímulos verbais utilizados são expandidos, ou seja, aumentados em sua duração, passando progressivamente para a duração normal da fala. Nesta estratégia a principal habilidade auditiva requisitada é a discriminação auditiva para sons verbais.

No quarto jogo, **Jogando com o Zé Jacaré,** a temática é um jacaré jogador de futebol, que é controlado pela criança. Dessa forma a criança tem que prestar atenção no estímulo que está sendo apresentado (o estímulo repete continuamente) e quando ouvir um estímulo diferente deve largar o mouse. Neste momento o animal chuta a bola. Se ela largar o mouse no momento certo marcará o gol, caso erre, a bola irá para fora. Os estímulos são verbais e os níveis de dificuldade da estratégia são determinados por encontros vocálicos, CVC e VCV com diferenças de ponto, modo e sonoridade dos fonemas. Nesta atividade está envolvida principalmente a habilidade de atenção sustentada. [7] definiram a atenção como sendo a *habilidade para persistir ouvindo durante um período de tempo, focalizando a audição para um estímulo predeterminado*. Envolve a monitorização do sinal acústico, priorizando-o em relação aos demais sinais competitivos a fim de que possa tornar-se significativo ([7]), mesmo que o sujeito necessite manter sua atenção primária em outra modalidade sensorial ([5]).

O último jogo denominado **Correndo com leão Léo**, tem como finalidade auxiliar a criança a desenvolver a atenção seletiva por meio de uma brincadeira em que a criança tem que detectar palavras, sentenças e rimas identificando-as nas opções apresentadas, isso em meio ao ruído competitivo. Estes estímulos verbais que são detectados tem variações de dificuldade caracterizados por diferenças de ponto, modo e sonoridade dos fonemas. A dificuldade de ruído competitivo segue a temática de que Léo é um corredor de rally e estará correndo por diferentes pistas e, dessa forma, esses ruídos são uma das dificuldades manipuladas pelo software e encontrada pela criança. Outra dificuldade estará na diminuição da expansão do estímulo. De acordo com [1] a atenção seletiva corresponde ao processo pelo qual selecionamos, ou seja, a capacidade de selecionar algumas informações para um exame mais detalhado e ignorar outras.

3. RESULTADOS

No momento presente, o desenvolvimento do SARDA encontra-se na fase final, e ao mesmo tempo estão sendo selecionadas as crianças que irão testar o sistema.

O projeto SARDA utilizou o Flash, o que possibilitou o desenvolvimento de jogos interativos, estimulando assim o interesse das crianças e retornando os dados referentes a interação das crianças com o sistema, para que os fonoaudiólogos possam estudar o comportamento auditivo dos pacientes. Desta forma, tornou-se necessário criar um sistema, invisível ao usuário, que trabalhe com as informações entre o jogo e o banco de dados.

O fluxo de informações entre o jogo e o banco de dados é bastante complexo. Para esta tarefa fundamental no projeto foi utilizado o PHP (Hipertext Preprocessor), que gerencia e direciona o curso de dados entre o aplicativo Flash e o banco de dados, sendo que as páginas em PHP têm apenas a função de transmitir os dados. A interface gráfica das páginas, que é visível ao usuário, foi criada em HTML (HyperText Markup Language). Utilizou-se o PHP porque é uma linguagem de programação livre e amplamente recomendada quando se trabalha com banco de dados.

Ao logar no site, a criança tem acesso a um ambiente onde pode acessar os jogos que poderá jogar (de acordo com a configuração pré-definida pelo fonoaudiólogo). Ao acessar o jogo, a criança tem acesso a uma tela de instruções (Figura 2) que contém informações referentes à como jogar.

Figura 2: Instruções do jogo “Localizando com o Mico Mané”

Neste momento, o arquivo em flash envia uma solicitação, através de uma camada intermediária desenvolvida em linguagem PHP, ao banco de dados, para obter informações que são utilizadas no jogo. Ao finalizar o jogo, informações referentes ao tempo utilizado para encerrar o nível, quantidade de acertos, níveis e jogos que a criança executou, bem como outras, são enviadas ao banco de dados, e assim o fonoaudiólogo, através de consultas, pode avaliar posteriormente o resultado. Todos os jogos desenvolvidos utilizam à mesma estrutura de interface, para que a criança se familiarize com o jogo.

A pergunta é: porque é necessário trabalhar com essas informações entre o jogo e o banco de dados? Porque o objetivo do projeto, não é apenas criar um jogo para o lazer. A meta a ser alcançada é criar um jogo capaz de recolher informações comportamentais do paciente em relação ao que o jogo exige dele, ou seja, é necessário que estas informações sejam gravadas em um banco de dados para serem analisadas futuramente por uma equipe de profissionais. Além disso, baseado no fato de cada criança possuir um comportamento diferente, criou-se

um sistema flexível que se adapte ao paciente. É por isso que foram utilizados elementos do banco de dados para que cada jogo seja personalizado de acordo com o usuário.

Quando o arquivo em Flash é iniciado, uma tela de apresentação surge no monitor. Ao lado direito inferior da janela, um botão "Continuar" é apresentado. Para o usuário, é apenas um simples botão. No entanto, ele apresenta uma função maior do que apenas seguir para a próxima tela. Ao clicar com o mouse sobre ele, um código programado em Flash ativa uma página em PHP. Essas operações são instantâneas e implícitas. A página em PHP contém códigos que buscam informações específicas no banco de dados e ficam temporariamente armazenas em algumas variáveis. Alguns exemplos de dados são: nome do jogador, quantidade de vezes que jogou aquela fase, se pode ou não jogar aquela fase específica, e outros dados referentes à configuração interna do jogo (limite de jogadas, tempo máximo, máximo de erros). Todas essas informações são depois enviadas pelo PHP ao jogo personalizado para o jogador.

No momento em que o jogo inicia, o usuário deve cumprir os objetivos para que possa completá-lo. Enquanto isso, o Flash vai guardando diversas informações que serão enviadas ao final. Estes dados podem ser, por exemplo, a pontuação, o tempo para completar o jogo, quantidade de acertos e de erros e outros dados que variam para cada atividade. Ao final do jogo, o Flash chama um outro arquivo em PHP que recebe estas estatísticas, e faz o envio delas para uma tabela do banco de dados. Ao enviar as informações, o PHP envia também a data em que o jogo foi acessado, e o identificador do usuário, um código que referencia os dados ao paciente. Futuramente, estes elementos poderão então ser analisados, para avaliação e aplicação de testes, em uma página restrita aos profissionais que irão utilizá-los.

Figura 3: Animação do jogo "Localizando com o Mico Mané"

A primeira estratégia desenvolvida foi "Localizando com o Mico Mané", que tem por objetivo fazer com que a criança perceba em que direção o som está saindo, ou seja, se está sendo tocado na orelha direita ou na esquerda. A cada acerto a criança ganha 100 pontos, e o personagem Mico Mané aparece em uma animação demonstrando que a criança acertou. Outras animações aparecem aleatoriamente durante o jogo - Figura 3, para estimular a criança a continuar interagindo com o jogo.

Para que uma fase seja considerada cumprida é necessário que a criança acerte 80% do jogo, e caso isto não ocorra ela tem mais uma oportunidade de jogar. Existem animações que vão ocorrendo conforme o jogo vai sendo executado, com o objetivo de motivar a criança a encerrar a fase. Ao final de cada fase o personagem do jogo ganha um presente, e ao término do jogo é exibido um pequeno filme a respeito do que aconteceu com o personagem. Como exemplo, pode-se citar o Mico Mané. Quando a criança passa de etapa, o mico recebe um presente. Ao passar da etapa fácil o mico ganha uma prancha de surf, na segunda recebe um carro, e na terceira o direito a ir à praia (Figura 4).

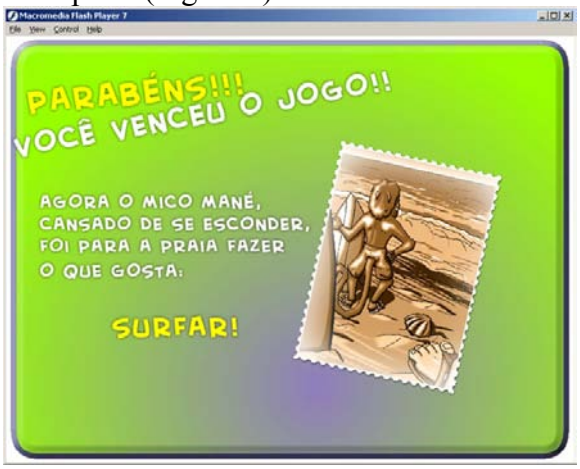


Figura 4: Tela final do jogo do Mico

Todos os outros jogos desenvolvidos seguem o mesmo padrão, e mudam o personagem, o ambiente e a estratégia utilizada.

O primeiro protótipo da estratégia "Localizando com o Mico Mané" foi aplicado em cinco crianças de seis a nove anos de idade. A experimentação foi fundamental para que elas respondessem a um questionário de oito perguntas para a validação da estratégia. No questionário constavam cinco modelos de respostas ilustradas em desenhos da face do mascote do SARDA, as faces correspondiam a notas de um a cinco. Na tabela 1 tem-se as perguntas aplicadas e a média das respostas das crianças:

Tabela 1. Perguntas aplicadas na estratégia "Localizando com o Mico Mané"

PERGUNTAS	MÉDIA DAS RESPOSTAS
Você entendeu as instruções?	3.8
Você achou fácil passar pelas instruções?	3.8
Você gostou das animações e dos desenhos?	4.2
Você entendeu bem como jogar o jogo?	4.4
Você conseguiu fazer o que dizia nas instruções?	4.2
Você entendeu para que serviam os botões?	5.0
Você gostou dos desenhos e animações?	4.8
Você achou divertido jogar?	3.8

Com a análise destes resultados pode-se concluir a necessidade de aperfeiçoar a apresentação de alguns quesitos da estratégia para que as crianças possam compreendê-las com maior facilidade. Por outro lado houve receptividade das crianças a interface do jogo evidenciando a necessidade de inserção de um maior número de animações ao longo da execução do jogo.

4. CONCLUSÕES

No Brasil ainda há necessidade de se ter recursos interativos e atraentes as crianças e que as auxiliem a desenvolver as habilidades auditivas como recurso terapêutico e como recurso educacional. Por vezes observam-se as professoras de pré-escola e primeiras séries escolares auxiliando pouco no desenvolvimento destas habilidades auditivas.

Este fator chega a ser um contra senso visto que as professoras requisitam da criança ao longo do período escolar cerca de 70 a 80% destas habilidades, isto é, a maior parte das informações transmitidas pelo professor em sala de aula é realizada por meio da voz falada com predominância de estímulos auditivos. Esses por sua vez, são temporários e abstratos e requerem desenvolvimento eficaz da audição, atenção, processamento e memória.

O SARDA está em fase final de desenvolvimento, e espera-se que com a sua utilização, fonoaudiólogos e professores possam trabalhar em conjunto para melhorar as habilidades auditivas de crianças em desenvolvimento atuando na promoção e prevenção da saúde auditiva e dos distúrbios auditivos contribuindo, conseqüentemnete, para minimizar as conseqüências destes distúrbios.

Desta forma, espera-se observar mudanças nas habilidades auditivas antes e após a intervenção utilizando o software; aspectos que serão mensurados na etapa de validação no contexto terapêutico e educacional.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Atkinson, R.L et al. *Introdução à psicologia de Hilgard*. Porto Alegre: Artmed, 2002.

[2] Boothroyd, A. *Hearing impairment in young children*. Englewood Cliffs: Prentice Hall Inc, 1982.

[3] IBGE. Censo demográfico. "*Características gerais da população*" (2003). Rio de Janeiro.

[4] Northern, J. L.e Downs, M.P (2005). "*Audição na infância*". 5.ed. Rio de Janeiro: Guanabara Koogan.

[5] Pereira, L.D. Processamento auditivo. *Temas sobre desenvolvimento*, v.2, n.11, 1993.

[6] Pupo, A. *Alguns aspectos do processo de discriminação auditiva de sons da fala em crianças*. 1981. Dissertação (Mestrado em Ciências) - Pontifícia Universidade Católica de São Paulo, São Paulo, 1981.

[7] Russo, I.C.P.; Behlau, M. *Percepção da fala: análise acústica do Português Brasileiro*. São Paulo: Lovise, 1993.

[8] Scientific Learning. "*Fast Forword*". (2005). Disponível em: http://www.scilearn.com. Acesso em: 10/06/2005.

[9] Temple. "*Neural deficits in children with dislexia ameliorated by behavioral remediation: Evidence from functional MRI*" (2003). Proceedings of the National Academy of Sciences of the United States of America, v. 100, n. 5, p.2860-2865.

Dimensión Fractal de Autocorrelación Cuadrática en Imágenes Digitales

Andrea Silvetti y Claudio Delrieux

Departamento de Ing. Eléctrica y Computadoras - Universidad Nacional del Sur - claudio@acm.org
- Parcialmente financiado por la SECyT-UNS

Abstract

El método de autocorrelación cuadrática es una variante para computar coeficiente de Hurst, el cual se utiliza para medir la dimensión fractal local en imágenes digitales. Usualmente, para computar dicho coeficiente, se toma una ventana cuadrada centrada en el pixel p, cuya dimensión fractal local se desea medir, y se evalúa la variancia de la luminancia ΔV entre dicho pixel y los pixels que están a una distancia r de p. La pendiente de la regresión lineal de dicha variancia en función de r en un espacio logarítmico es una estimación del coeficiente de autocorrelación H local en p, y la dimensión fractal local es $3 - H$. La autocorrelación cuadrática, en cambio, evalúa la variancia de luminancia entre *todos* los pixels dentro de la ventana para una distancia dada. Por lo tanto, la regresión de la variancia se realiza en función del área de dicha ventana. En trabajos anteriores mostramos que dicha evaluación es mucho más precisa y estable, pero con un costo computacional mucho mayor.

En este trabajo proponemos una mejora al método de autocorrelación cuadrática, la cual no solo reduce el costo computacional a menos de la mitad, sino que mejora experimentalmente los resultados obtenidos. Aún trabajando con ventanas pequeñas, se obtienen estimaciones más exactas y precisas con respecto a los métodos tradicionales con grandes ventanas, por lo cual los resultados son altamente satisfactorios. Además, mostramos que las propiedades de invariancia a transformaciones afines de geometría y luminancia del método son superiores al método tradicional. Finalmente, se muestran algunos ejemplos del uso de nuestro método en segmentación de características en imágenes médicas y de sensado remoto.

Palabras Clave: Procesamiento de Imágenes --- Dimensión Fractal --- Coeficiente de Hurst

1. Introducción

La estimación de la dimensión fractal (DF) en superficies es una técnica muy usada en procesamiento de imágenes, especialmente para la segmentación de regiones. Esto se debe fundamentalmente a la robustez que la caracteriza frente a los defectos y alteraciones presentes en la mayor parte de las tecnologías de adquisición de imágenes. En numerosas aplicaciones, los métodos tradicionales de segmentación no producen resultados satisfactorios, y por ello los estimadores fractales constituyen un adecuado descriptor local para segmentar regiones. Tal es

el caso del procesamiento de imágenes de sensado remoto (satélites ópticos y SAR), imágenes médicas (PET, CAT, ultrasonido), visión robótica e industrial, etc. [10, 1, 4].

La DF es un exponente que relaciona la autoinvariancia afín o autosimilitud estadística frente a cambios de escala [3, 5]. En conjuntos matemáticamente definidos, es posible encontrar la DF en forma determinística utilizando la DF de autosimilitud (llamada también DF de Hausdorff, aunque la propuesta original se debe a Kolmogorov). Sin embargo, este método es inadecuado cuando no existe una definición matemática del conjunto. Por dicha razón, en objetos o modelos no determinísticos, como puede ser la función de luminancia de una imagen digital, la metodología usual para obtener dicho exponente consiste en realizar el ajuste por regresión, (en un espacio logarítmico), de alguna magnitud relevante del conjunto, como función inversamente proporcional a la escala de medición de dicha magnitud. En algunos casos, lo que se mide es una magnitud indirecta (ocupación, probabilidad, entropía, amplitud espectral), y por lo tanto se utilizan dimensiones fractales como la DF Box, la DF de entropía, o la DF de Fourier. En el caso de imágenes digitales, la variación de luminancia es una magnitud de obtención inmediata, y por lo tanto es adecuado computar la DF local basada en las variaciones locales de luminancia [7, 6].

El cómputo de la DF local en un pixel p requiere realizar la regresión del comportamiento de la luminancia en función de la escala, en un entorno reducido centrado en p. En imágenes digitales, lo usual es que dicho entorno esté incluido en una *ventana* cuadrada. En general, cuanto mayor es la ventana, más precisa es la estimación pero mayor es el costo computacional. Uno de los métodos más usuales para evaluar la DF local es el exponente de autocorrelación de Hurst (H), que caracteriza a una función de acuerdo a cómo se puede estimar su variancia ΔV dentro de entornos r de tamaño decreciente, de manera de obtener un ajuste de la forma $\Delta V \sim (\Delta r)^H$. Es posible mostrar que la dimensión fractal de la función es $D = 2 - H$ con $H = \lim_{r \to 0} \frac{\log(\Delta V)}{\log(r)}$ y $0 \leq H \leq 1$.

En imágenes, se toman discos de radios decrecientes centrados en el pixel cuya DF local se desea medir, y se evalúa la regresión del ΔV dentro del disco, en función del radio r. La relación entre la dimensión fractal de la función de luminancia de la imagen y el coeficiente de autocorrelación es $D = 3 - H$ dado que H se computa con la regresión de $H = \lim_{r \to 0} \frac{\log(\Delta V)}{\log(r)}$.

En [8] se introduce una variación al método de cálculo del coeficiente de autocorrelación adaptado a superficies. Este nuevo método evlúa un *coeficiente de autocorrelación cuadrático* mejorando la calidad de las estimaciones locales de la DF, pero con un costo computacional mucho más alto a iguales tamaños de ventana. En este trabajo se presenta un forma de evaluar dicho coeficiente reduciendo la complejidad de cálculo, obteniendo resultados igual de precisos. Estudiamos también las propiedades de invariancia frente a transformaciones afines de geometría y luminancia de esta nueva técnica. Comparamos nuestro método con las implementaciones usuales en la bibliografía, y mostramos que la DF local obtenida con el coeficiente de autocorrelación cuadrático evaluado en ventanas pequeñas es más precisa que la obtenida con el método tradicional en ventanas de mayor tamaño. Finalmente, mostramos algunos ejemplos de aplicación de la DF local para la segmentación de características en imágenes de microscopía electrónica y de sensado remoto.

2. Autocorrelación cuadrática de la luminancia

Las técnicas para medir la DF local en superficies mencionadas en este trabajo están basadas en la determinación del exponente de autocorrelación de la luminancia en imágenes. El método

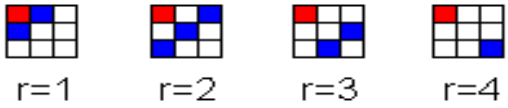


Figura 1: Pixels a distancia 1,2,3 y 4 (en norma 1) respecto del pixel superior izquierdo

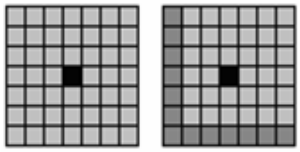


Figura 2: Ventana de 7×7, mostrando que 12 es la máxima distancia en norma uno.

tradicional para el cómputo de la DF de autocorrelación evalúa la regresión de la variancia de luminancia ΔV en función del radio r del entorno considerado, tomando valores decrecientes del radio dentro de la ventana. En la determinación de ΔV, un incremento Δr del radio del disco incrementa en $2\pi\Delta r$ la cantidad de pixels a evaluar. Por ello decimos que es un método 1D de revolución, y en consecuencia, aún trabajando con ventanas de gran tamaño, los tiempos de ejecución son relativamente cortos, pero los resultados no son muy satisfactorios.

En [8] tomamos las áreas dentro de cada ventana precisamente como *áreas rectangulares* y no como círculos de radio r, y por lo tanto en la determinación de ΔV se consideran *todos* los pixels que están a una distancia r entre sí dentro de la ventana. Se desplaza una ventana de $n \times n$, n impar, sobre cada pixel p de la imagen para calcular la DF local del mismo. Se busca la máxima variación de luminancia $\Delta V = |L_i - L_j|$ entre dos pixels i y j dentro la ventana tales que i y j están a distancia r. Utilizamos distancia norma uno, y por lo tanto se cosidera la pendiente de la regresión en espacio logarítmico entre ΔV y r, para valores de r entre 1 y $2n-2$ que es la máxima distancia norma uno entre dos pixels cualesquiera de una ventana cuadrada de $n \times n$ (ver Fig. 1). La regresión buscada es ahora entre la variancia de luminancia ΔV y el área rectangular ar considerada. El coeficiente de autocorrelación cuadrático H_2 es entonces

$$H_2 = \lim_{ar \to 0} \frac{\log(\Delta V)}{\log(ar)} = \lim_{r \to 0} \frac{\log(\Delta V)}{\log(r^2)} = \lim_{r \to 0} \frac{\log(\Delta V)}{2\log(r)}.$$

El uso de distancia norma uno permite que, aún con ventanas pequeñas, la cantidad de puntos en la regresión sea mucho mayor (ver Fig. 2).

Teniendo en cuenta el tiempo de cómputo para la obtención del coeficiente H_2, estudiamos la aplicación del método en áreas circulares, lo cual determina una disminución de la cantidad de pixels considerados y consecuentemente del tiempo de ejecución. En esta mejora, se desplaza una ventana circular de diámetro n impar sobre cada pixel de la imagen para calcular la DF local del mismo. De este modo, hacemos la regresión entre ΔV y r, para valores de r entre 1 y $2 \times \text{máx}(i+j)$ para i, j tales que $i^2 + j^2 \leq (\lfloor n/2 \rfloor)^2$ (ver Fig. 3). De esa manera, la correlación entre ΔV y el área circular ac es

$$H_2 = \lim_{ac \to 0} \frac{\log(\Delta V)}{\log(ac)} = \lim_{r \to 0} \frac{\log(\Delta V)}{\log(\pi r^2)} = \lim_{r \to 0} \frac{\log(\Delta V)}{\log(\pi) + 2\log(r)}.$$

Es posible ver que, en el espacio logarítmico, la regresión de puntos ΔV *vs.* ac tendrá una pendiente igual a $2H$, dado que el término constante en el denominador implica un desplazamiento de todos los puntos sin alterar la pendiente del ajuste.

En otras palabras, la ventana circular es también un método de autocorrelación cuadrática, pero con $\pi/4$ veces menos pixels a considerar, mejorando notoriamente el tiempo de ejecución.

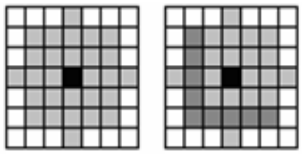


Figura 3: Ventana circular de diámetro 7, donde 8 es la máxima distancia en norma uno.

Método de medición	Tamaño de ventana				
	3	5	7	9	11
AL	8	20	36	68	80
AR	43	440	1911	5592	13095
AC	43	89	580	1692	5087

Tabla 1: Cantidad de pares de puntos a evaluar para determinar la autocorrelación local en un pixel, en función del tamaño de la ventana. AL: Autocorrelación lineal. AR: Autocorrelación cuadrática en áreas rectangulares. AC: idem AR en áreas circulares.

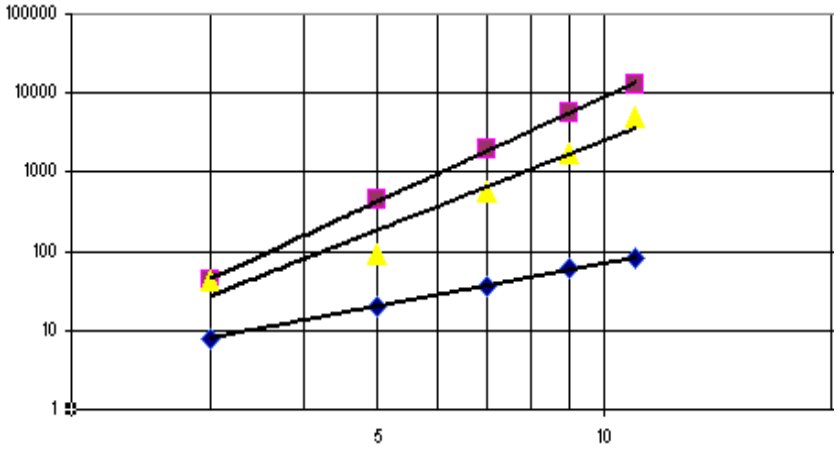


Figura 4: Regresión polinomial de la complejidad de los métodos.

En la Tabla 1 se puede apreciar la cantidad de pares de puntos a evaluar para determinar la autocorrelación local en un pixel, en función del tamaño de la ventana. En la Fig. 4 se visualiza la regresión polinomial de la cantidad de comparaciones de los tres métodos en función del tamaño de la ventana. Las pendientes del ajuste determinan que la complejidad del método tradicional (lineal) es $O(N^{1,83})$, el rectangular es $O(N^{4,40})$, y el circular $O(N^{3,37})$. Por otra parte, como veremos en la próxima Sección, esta mejora produce resultados de idéntica calidad que la estimación cuadrática, pero con una apariencia visual más adecuada, especialmente en las propiedades de invariancia frente a transformaciones.

Para efectuar la comparación de nuestro método con el tradicional, tomamos una imagen sintética con 6 sectores y un borde, cada uno con diferente DF local. Estos sectores fueron generados utilizando el algoritmo de desplazamiento aleatorio del punto medio [2] (ver Fig. 5). Los resultados de la medición de la DF con la técnica de autocorrelación cuadrática son muy superiores a los obtenidos con el método de Hurst tradicional (ver Figs. 6). La dimensión fractal local se muestra en falso color, con escala variando entre D=2 (verde) y D=3 (rojo). En la Tabla 2 se consigna el promedio de las DF locales en los pixels de cada sector, medidas con el coeficiente de autocorrelación lineal de Hurst (AL), con el coeficiente de autocorrelación

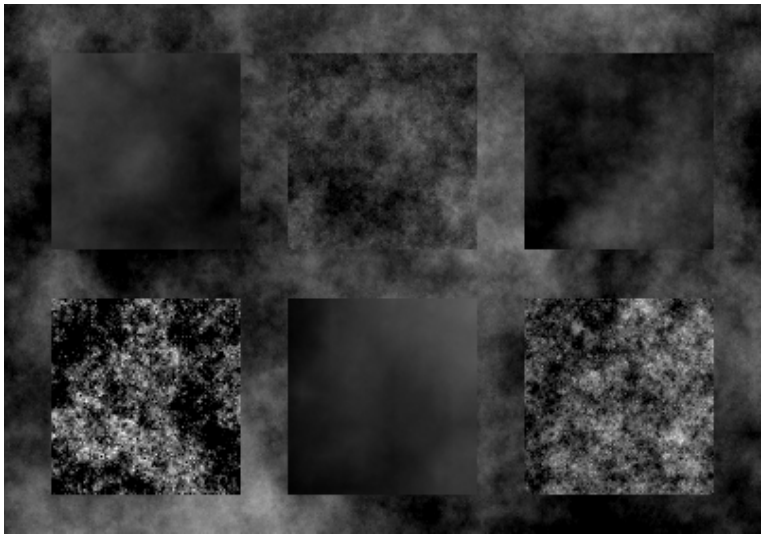


Figura 5: Imagen de prueba con 6 sectores y un borde, cada uno con diferente dimensión fractal.

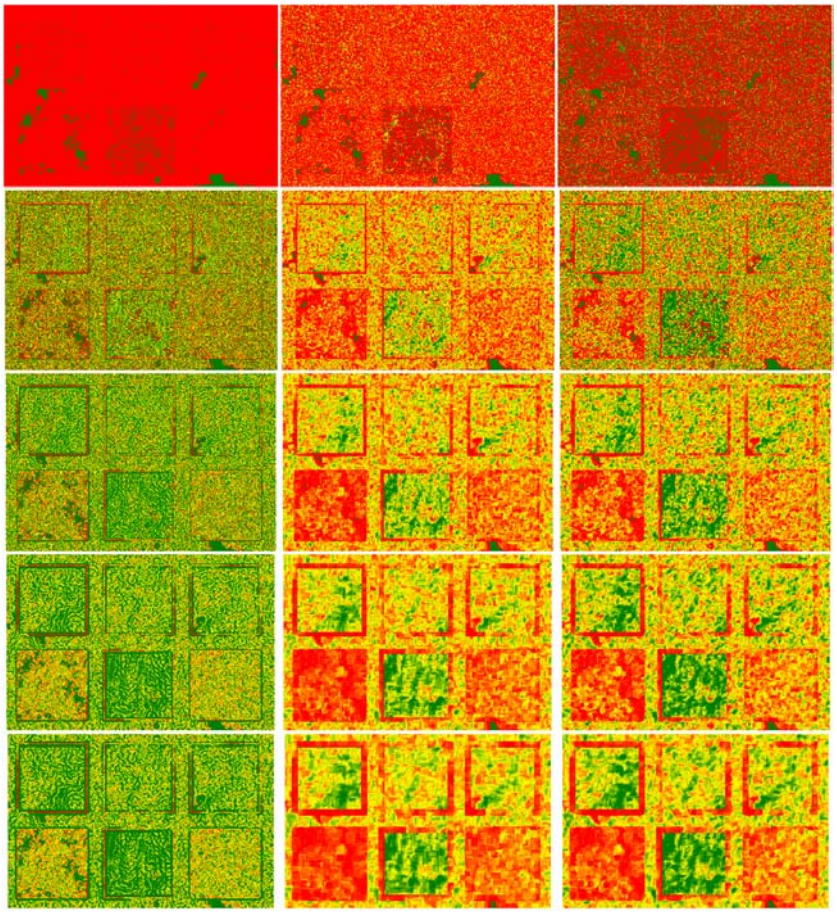


Figura 6: DF local computada a partir de la imagen de la Fig. 5 De izquierda a derecha: Métodos AL, AR y AC. De arriba hacia abajo: Tamaño de ventana 3, 5, 7, 9 y 11. La dimensión fractal local se muestra en falso color, con escala variando entre D=2 (verde) y D=3 (rojo)

cuadrático en áreas rectangulares (AR) y ciruclares (AC), en cada uno de los sectores, junto con la DF original del sector, para diferentes tamaños de ventanas. También se visualizan los mismos resultados en la Fig. 7 donde claramente se advierte que realizar una medición con

Sector		Ventana 3			Ventana 5			Ventana 7		
Ubicación	DF	AL	AR	AC	AL	AR	AC	AL	AR	AC
Sup. Izq.	2,2	2,9362	2,8205	2,7401	2,3762	2,5273	2,3848	2,2402	2,4418	2,3725
Sup.Cen.	2,6	2,9973	2,8754	2,7362	2,4336	2,72	2,5865	2,3435	2,6924	2,6337
Sup.Der.	2,4	2,9739	2,8365	2,7173	2,4002	2,6223	2,479	2,2863	2,5645	2,5005
Inf. Izq.	2,9	2,904	2,8387	2,6944	2,4647	2,8385	2,7218	2,3972	2,8604	2,8134
Inf. Cen.	2,1	2,8875	2,7965	2,7119	2,3518	2,464	2,3138	2,195	2,3608	2,2867
Inf. Der.	2,8	2,9946	2,876	2,733	2,4527	2,7629	2,6322	2,3797	2,7515	2,6956
Borde	2,5	2,9953	2,8628	2,7211	2,42	2,6628	2,5177	2,3128	2,6125	2,5492

Sector		Ventana 9			Ventana 11		
Ubicación	DF	AL	AR	AC	AL	AR	AC
Sup. Izq.	2,2	2,2103	2,4041	2,3283	2,1863	2,3854	2,3323
Sup.Cen.	2,6	2,3433	2,683	2,6193	2,3263	2,6831	2,6366
Sup.Der.	2,4	2,2712	2,5387	2,4647	2,2489	2,5271	2,4733
Inf. Izq.	2,9	2,4153	2,8778	2,8303	2,4125	2,8935	2,8621
Inf. Cen.	2,1	2,1603	2,3188	2,2357	2,1386	2,2982	2,24
Inf. Der.	2,8	2,3919	2,756	2,6951	2,3819	2,764	2,7215
Borde	2,5	2,3018	2,591	2,5192	2,2789	2,5814	2,5296

Tabla 2: Resultados obtenidos con la imagen de prueba en la Fig. 5 para los tres métodos y difetentes tamaños de ventana

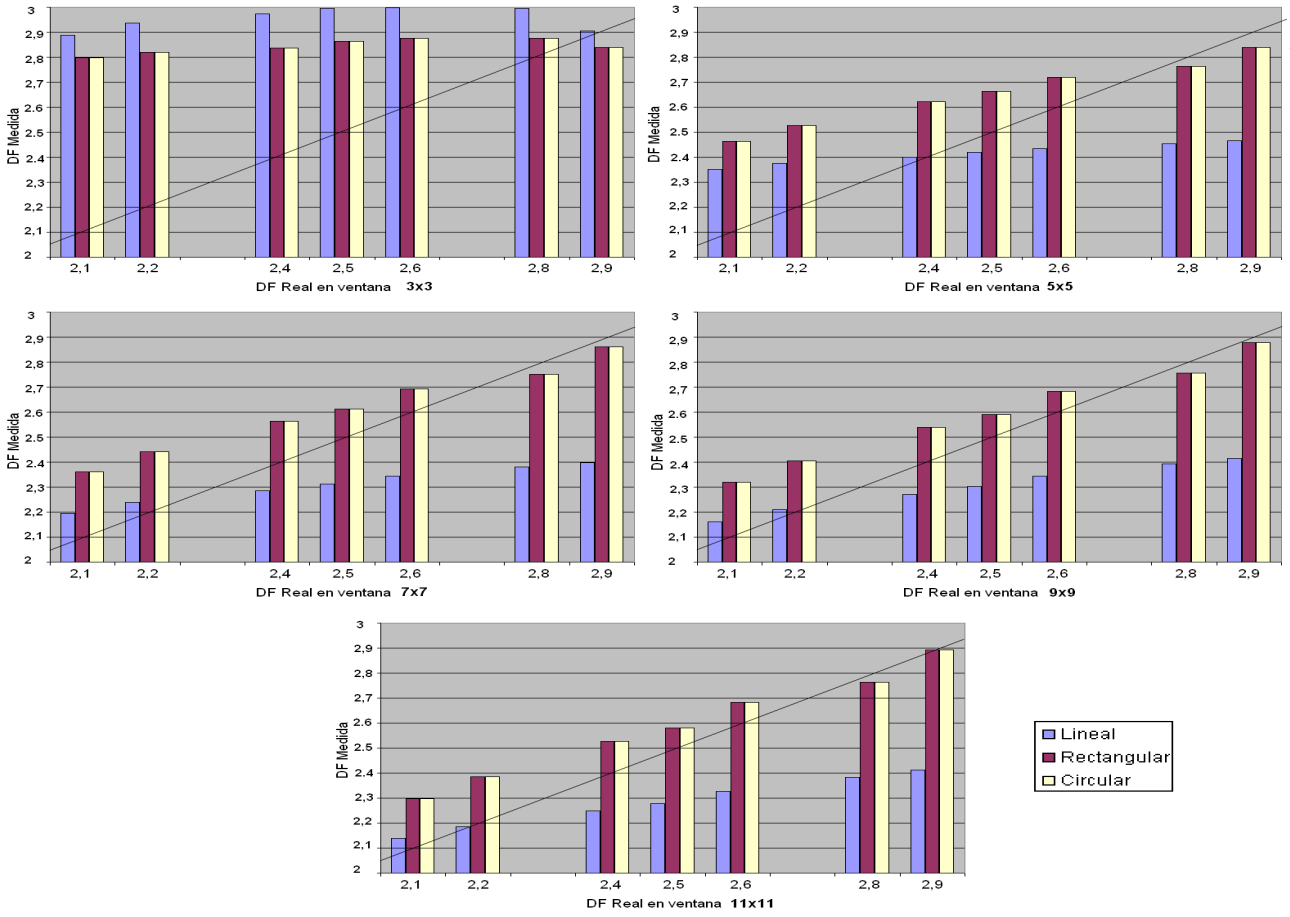


Figura 7: Mediciones obtenidas con la imagen de prueba en la Fig. 5 para los tres métodos y difetentes tamaños de ventana

ventana 9x9 con le método lineal es comparable con una medición con ventana 5x5 con el método circular.

3. Propiedades de invariancia

Testeamos el método de autocorrelación cuadrático con la nueva mejora frente a un conjunto de propiedades que debe tener para ser considerado adecuado como método de medida [9]. En particular, la propiedad de invariancia de una medida W es denominada *dimensional* si posee la propiedad

$$W(\lambda_g \lambda_l f) = W(f),$$

donde $\lambda_g \lambda_l f$ denota la aplicación de una transformación geométrica afín g y una transformación afín de luminancia l a la función f de luminancia de la imagen. Esta propiedad básicamente establece que la medición local de la DF debe ser invariante frente a transformaciones afines de la geometría y de la luminancia, y es fundamental para asegurar que la DF es una medida robusta. En todos los casos, comparamos resultados sobre los testeos realizados tanto en los métodos de autocorrelación cuadrático en áreaas rectangulares (AR) y circulares (AC), como en el método de autocorrelación lineal tradicional (AL).

Es fácil mostrar que en el caso de traslaciones, la variancia de luminancia ΔV no experimenta cambios simplemente por definición. Dada una traslación t, $|L_{i+t} - L_{j+t}|$ será idéntica a $|L_i - L_j|$ para cualquier par de pixels; y esto vale para cualquiera de los métodos analizados. En el caso de invariancia frente a rotaciones, es esperable que ocurran pequeñas diferencias producidas por el *aliasing* y filtrados implícitos de remuestreo y reconstrucción que generan las rotaciones en una grilla discreta. En la Fig. 8 se observa el resultado de aplicar DF local a la imagen de prueba de la Fig. 5 con autocorrelación lineal y autocorrelación cuadrática en áreas circulares, para ventana de tamaño 11, y con diferentes ángulos de rotación. En la Tabla 3 se consignan los resultados con los tres métodos de medida.

La invariancia frente a cambios de escala también es teóricamente esperable, dado que aplicar una escala isotrópica α a la imagen equivale a modificar el radio del entorno r por su factor recíproco. Por lo tanto la definición de H es ahora

$$H = \lim_{r \to 0} \frac{\log(\Delta V)}{\log(\frac{r}{\alpha})},$$

lo que implica que los puntos en el ajuste por regresión se desplazan horizontalmente un valor constante, pero la pendiente del ajuste sigue siendo igual. Sin embargo en la práctica, el cambio de escala de una imagen implica su remuestreo y reconstrucción, lo cual genera *aliasing* y filtrados implícitos especialmente en las frecuencias más altas (es decir, el ΔV medido para valores de r bajos está por debajo del real). En este caso es donde nuestro método exhibe mayor robustez que el tradicional, dado que cuenta con mayor cantidad de puntos en la regresión y por lo tanto la subdeterminación en valores de r bajos no afecta tan severamente al H obtenido. En la Fig. 9 se observa el resultado de aplicar DF local a la imagen de la Fig. 5 con diferentes escalas procesadas con ventana de tamaño 7. En la Tabla 4 se consignan los resultados.

Con respecto a transformaciones afines de luminancia, es fácil ver que en teoría los métodos deberían ser invariantes. Supongamos una transformación de brillo y contraste

$$L' = \beta L + \gamma.$$

La traslación por γ claramente no afecta al nuevo ΔV, dado que se cancela en la diferencia de valores de luminancia entre todo par de pixels. El factor de escala β va a aparecer como factor

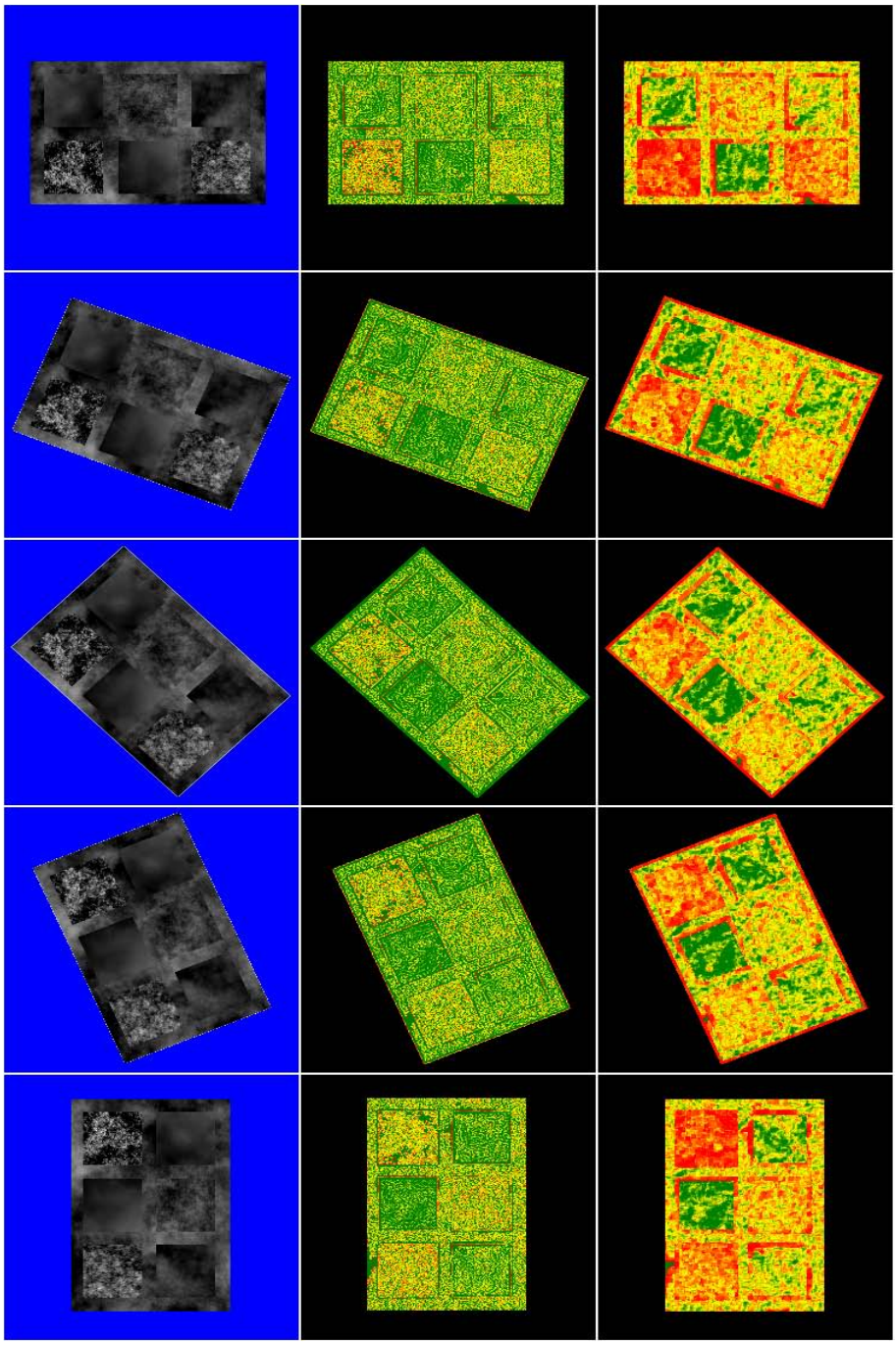


Figura 8: Invariancia frente a rotaciones. Columna de la izquierda: Imagen de prueba rotadas 0, 22, 45, 67 y 90 grados. Columna del centro: AL. Columna de la derecha: AC.

Invariancia frente a rotaciones					
Método	0º	22º	45º	67º	90º
AL	2.2751	2.2347	2.2248	2.2360	2.2780
AR	2.5772	2.5709	2.6187	2.5767	2.6189
AC	2.5693	2.5032	2.4907	2.5017	2.5701

Tabla 3: DF local media en las imágenes de la Fig. 8. Ventana de tamaño 11.

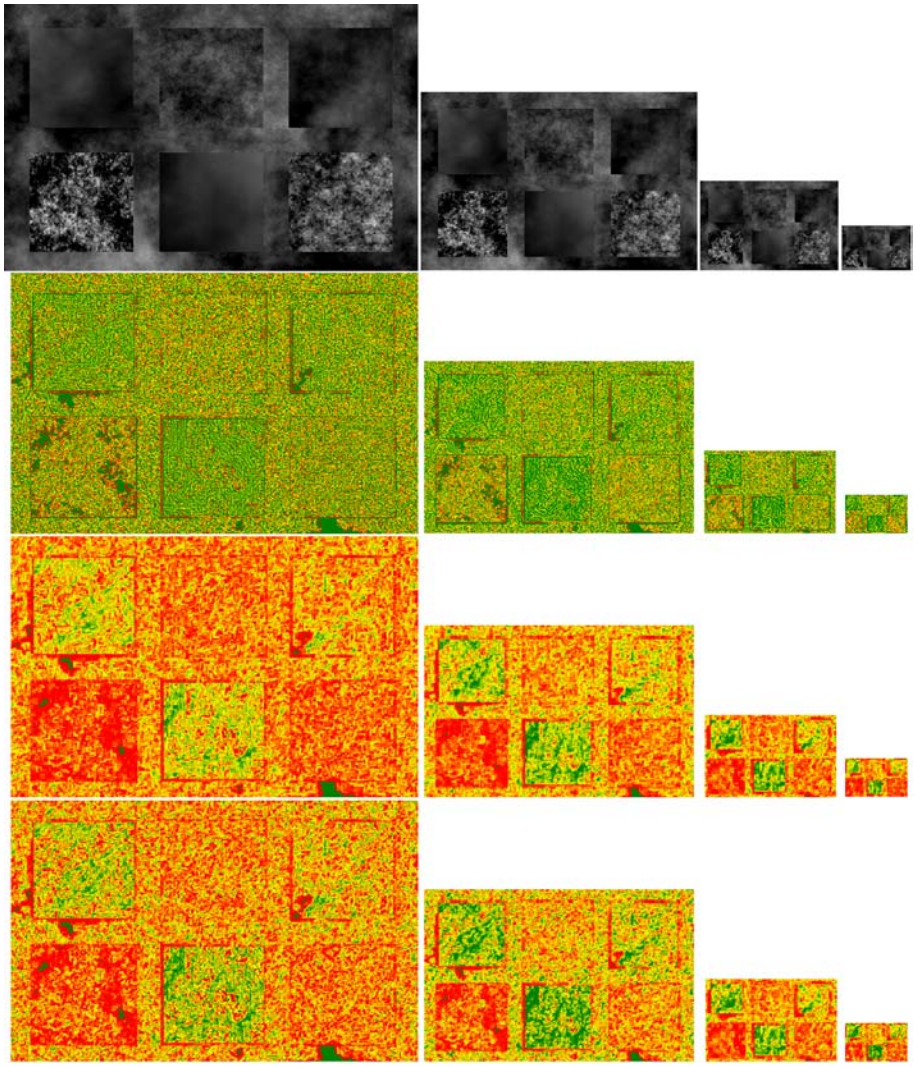


Figura 9: Invariancia frente a cambios de escala sobre la imagen de la Fig. 5. Fila superior: Transformación de escala al 150 %, 100 %, 50 % y 25 %. Filas siguientes: procesamiento con AL, AR y AC respectivamente

del nuevo ΔV ya que afecta a la luminancia de ambos pixels. Por lo tanto en la regresión lineal los valores de $\log(\Delta V)$ aparecerán desplazados por un factor constante, lo cual implica que su pendiente tampoco es afectada. En la práctica, como L está representada en 8 bits, es muy frecuente que los efectos de cuantización generen valores espurios de DF. Como en las simetrías anteriores, nuestro método es más robusto que el tradicional con respecto a transformaciones de luminancia. En la Fig. 10 se observa el resultado de aplicar DF local a la imagen de a Fig. 5 con varias transformaciones de luminancia y ventana 7. En la Tabla 5 se consignan los resultados.

Invariancia frente a cambios de escala												
150 %			100 %			50 %			25 %			
AL	AR	AC	AL	AR	AC	AL	AR	AC	AL	AR	AC
2.3489	2.7035	2.6526	2.3028	2.6288	2.5651	2.3132	2.6458	2.5847	2.3139	2.6863	2.6230

Tabla 4: DF local media en las imágenes de la Fig. 9. Ventana de tamaño 7.

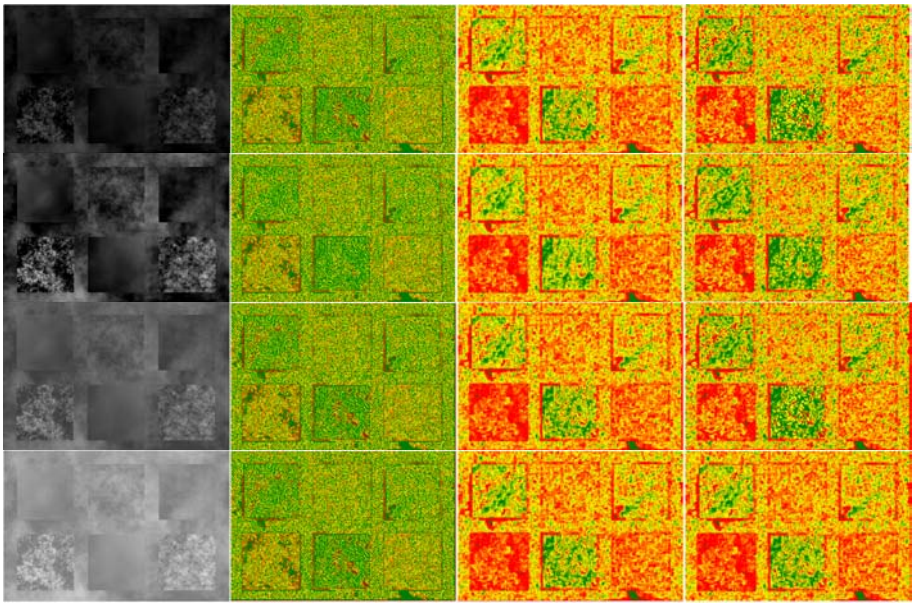


Figura 10: Invariancia frente a cambios de luminancia. En la columna de la izquierda se aplicaron transformaciones de lumninancia: $L = 0{,}5L$, $L = L$ (luminancias de la imagen original, en el intervalo [0..255]), $L = 64 + 0{,}5L$ y $L = 127 + 0{,}5L$. Las demás columnas de izquierda a derecha corresponden al procesamiento con ventana de tamaño 7 con AL, AR y AC respectivamente.

Invariancia frente a transformaciones de luminancia												
DF del sector	L original			$L = 0{,}5 * L$			$L = 64 + 0{,}5 * L$			$L = 127 + 0{,}5 * L$		
	AL	AR	AC	AL	AR	AC	AL	AR	AC	AL	AR	AC
2,2	2,2328	2,3495	2,4216	2,2423	2,477	2,4194	2,2423	2,477	2,4194	2,2423	2,477	2,4194
2,6	2,3391	2,6228	2,6824	2,3246	2,6947	2,6358	2,3246	2,6947	2,6358	2,3248	2,6947	2,6358
2,4	2,2799	2,4857	2,5518	2,2717	2,5745	2,5129	2,2717	2,5745	2,5129	2,2717	2,5745	2,5129
2,9	2,3932	2,8104	2,8582	2,3852	2,8617	2,815	2,3854	2,8617	2,815	2,3875	2,8617	2,815
2,1	2,186	2, 2622	2,3391	2,2225	2,3877	2,3223	2,2225	2,3877	2,3223	2,2225	2,3877	2,3223
2,8	2,3757	2,6903	2,7468	2,3635	2,7538	2,6983	2,3638	2,7538	2,6983	2,3665	2,7538	2,6983
2,5	2,3075	2,533	2,5995	2,2945	2,6148	2,5522	2,2945	2,6148	2,5522	2,2971	2,6148	2,5522

Tabla 5: DF local media en las imágenes de la Fig. 10. Tamaño de ventana 11.

4. Resultados y aplicaciones

En esta sección mostramos algunos resultados obtenidos al aplicar nuestro método en un par de ejemplos de imágenes de diferentes fuentes. En la Fig. 11(a) podemos observar una imagen resultante de promediar un conjunto de imágenes satelitales nocturnas NOAA DMSP obtenidas sobre la provincia de Buenos Aires durante 2003[1]. El objetivo de obtener estas imágenes consiste en poder determinar los patrones de urbanización y crecimiento urbano en la región. En las zonas altamente urbanizadas, la iluminación nocturna es suficiente como para saturar el sensor y obtener el máximo registro en la lectura. En las zonas despobladas se produce el efecto contrario. Las zonas en las cuales se está dando una situación de transición son aquellas donde la iluminación es intermedia. Sin embargo, dado que la calibración del sensor es adaptativa, no existe un registro que permita determinar con exactitud el valor preciso de luminancia en cada pixel, por lo tanto segmentar las áreas geográficas de transición utilizando la luminancia pro-

[1]Imágen y procesamiento efectuado en el NOAA's National Geophysical Data Center, Earth Observation Group (`http://www.ngdc.noaa.gov/dmsp`). Cortesía de Sharolyn Anderson y Benjamin Tuttle, de la University of Denver y el National Ocean and Atmosphere Agency, respectivamente.

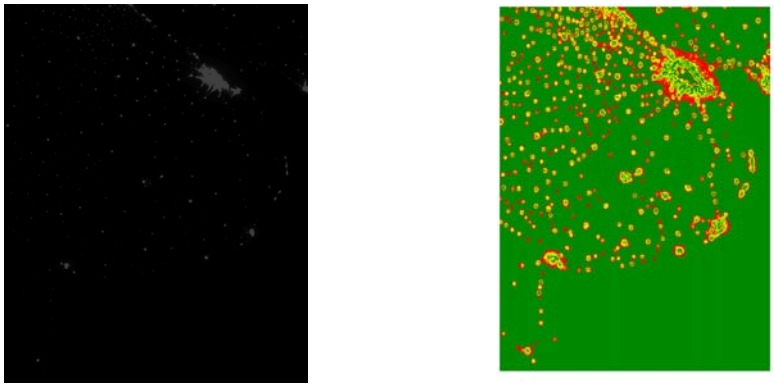


Figura 11: Imagen satelital nocturna y su dimensión fractal local.

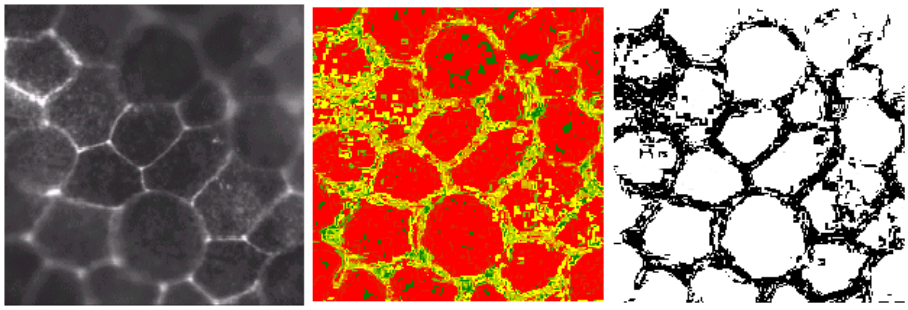


Figura 12: Imagen de microscopía electrónica, su dimensión fractal, y la binarización de los pixels con $DF < 2{,}8$.

duce resultados incorrectos. Si estos valores se utilizan para elaborar tendencias, los resultados pueden ser inconsistentes.

La caracterización fractal de la función de luminancia, sin embargo, no tiene dicho problema dado que es invariante frente a cambios de luminancia, y por lo tanto no depende de la calibración del sensor. Las zonas altamente urbanizadas, así como las zonas despobladas, se pueden caracterizar como de baja variancia geográfica, y por lo tanto de baja DF. Las zonas de transición serán aquellas en las cuales existe una baja o nula correlación geográfica en la función de luminancia, lo cual produce una alta DF (ver Fig. 11(b)).

En la Fig. 12(a) podemos ver una imagen del epitelio de un embrión de *Bufo arenarum* (sapo) obtenidos por técnicas de microscopía de inmunofluorescencia[2]. El propósito de este tipo de imágenes es el estudio de modelos biológicos, para lo cual una de las características fundamentales a determinar es la posición de la membrana celular. Este tipo de imágenes presenta una gran cantidad de problemas y defectos (ruido multiplicativo causado por el método de generación, desenfoque, cambios de contraste y luminancia debidos a la fijación no uniforme del reactivo, etc.). Todo esto hace imposible segmentar el contorno de dicha membrana con ninguno de los métodos clásicos de detección de bordes (filtrado por convolución, morfología, etc.). Sin embargo, una característica de la imagen es que en las zonas cercanas a las membranas celulares la intensidad de los pixels se hace más correlacionada que en otros lugares de la muestra, y por lo tanto su DF se hace más baja. Por lo tanto, una segmentación de las zonas de la imagen con DF menores que cierto umbral produce resultados que con posterior procesamiento pueden producir el resultado deseado (ver Fig. 12(c)).

[2]Cortesía Víctor Bravo y María Fernanda Izaguirre, Laboratorio de Microscopía, UNER.

5. Conclusiones y trabajo futuro

Presentamos una mejora al método de autocorrelación cuadrática para determinar el exponente de Hurst en superficies. La idea subyacente de este método consiste en evaluar la variancia de la luminancia dentro de la ventana, en todos los pixels que estén a cierta distancia r, y correlacionar dicha variancia con r. Se mostró que utilizando ventanas redondas es posible obtener mejores estimaciones locales de la dimensión fractal, y con un costo computacional menor. Se estudiaron las propiedades de invariancia frente a transformaciones afines de la geometría y de la luminancia y se compararon los resultados con las implementaciones usuales en la bibliografía Finalmente, mostramos algunos ejemplos de aplicación de la DF local para la segmentación de características en imágenes de microscopía electrónica, y de sensado remoto.

El trabajo futuro más importante es el estudio de la complejidad computacional del método y, fundamentalmente, encontrar mecanismos de aceleración que permitan su implementación en grandes volúmenes de información (imágenes 2D y datos 3D). Una estrategia posible para ésto, dada la natural independencia del cómputo local de la DF, consiste en utilizar hardware gráfico paralelo (GPUs).

Referencias

[1] C. Delrieux and R. Katz. Image Segmentation Through Automatic Fractal Dimension Classification. In *Argentine Symposium on Computing Technology*, Buenos Aires, 2003. 32 JAIIO, Jornadas Argentinas de Informática e Investigación Operativa,.

[2] A. Fournier, D. Fussell, and L. Carpenter. Computer Rendering of Stochastic Models. *Communications of the ACM*, 25(6):371–384, 1982.

[3] B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, New York, 1983.

[4] Yamashiro P. Fractal Analysis of MRI. Technical Report Medical Sciences Lab., University of Washington, Seattle, 1997.

[5] H.-O. Peitgen and D. Saupe. *The Science of Fractal Images*. Springer-Verlag, New York, 1986.

[6] J. C. Russ. *Fractal Images*. Plenum Press, 1993.

[7] J. C. Russ. *The Image Processing Handbook*. CRC Press, Boca Raton, FL, third edition, 1999.

[8] A. Silvetti and C. Delrieux. Medición de la Dimensión Fractal en Superficies. In *Workshop de Investigadores en Ciencias de la Computación*, pages 81–85, Trelew, Argentina, 2007. WICC, .

[9] Pierre Soille and Jean-F. Rivest. On the validity of fractal dimension measurements in image analysis. *J. Visual Communication and Image Representation*, 7(3):217–229, 1996.

[10] T. Wang, D. Li, C. Q. Zheng, and Y. Zheng. Segmentation of medical ultrasonic image using fractal. In J. Zhou, A. K. Jain, T. Zhang, Y. Zhu, M. Ding, and J. Liu, editors, *Proc. SPIE Vol. 3545, p. 496-499, International Symposium on Multispectral Image Processing (ISMIP'98), Ji Zhou; Anil K. Jain; Tianxu Zhang; Yaoting Zhu; Mingyue Ding; Jianguo Liu; Eds.*, pages 496–499, September 1998.

Generación de invariantes para implementar eficientemente Regiones Críticas Condicionales

Damián Barsotti
Javier O. Blanco
Fa.M.A.F., Universidad Nacional de Córdoba
Ciudad Universitaria, 5000 Cordoba, Argentina
{damian,blanco}@famaf.unc.edu.ar

25 de julio de 2007

Resumen

La técnica de semáforos binarios divididos (SBS) puede ser usada para implementar regiones críticas condicionales. Dada una especificación de un problema de esta clase, SBS brinda tanto los programas que lo implementan como los invariantes que aseguran su corrección. Aplicando la técnica a casos particulares se encuentran programas que admiten simplificaciones.

Este trabajo se concentra en el desarrollo de un procedimiento automático para obtener estas simplificaciones. El procedimiento consiste en hacer una búsqueda de nuevos invariantes que avalen la corrección de las simplificaciones. Para esto usamos técnicas de generación de invariantes, en particular propagación hacia atrás de las precondiciones más débiles. Su implementación fue realizada utilizando los demostradores Isabelle/HOL y CVC Lite para las demostraciones de validez y simplificaciones de las fórmulas lógicas envueltas en el proceso. El método fue probado sobre diferentes ejemplos clásicos de programación concurrente.

Abstract

Split binary semaphores (SBS) can be used to implement conditional critical regions. Given a specification of a program which use these constructs, the SBS technique produces not only the programs which implement the regions but the invariants which ensure their correctness as well. In most examples the programs obtained are suitable for further simplifications.

This work focus on the development of an automatic procedure to obtain these simplifications. We search for new (stronger) invariants which ensure correctness of the simplifications. For this aim, we use automatic invariant generation, in particular back-propagation techniques for weakest preconditions. Its implementation was developed using the Isabelle/HOL and CVC Lite provers for checking correctness and simplify logical formulae involved in the procedure. This method is illustrated on many classical concurrent programming examples.

Keywords: Programación Concurrente, Theorem Prover, SMT Solver, Semáforos Binarios Divididos, Invariantes, Lenguajes de Programación.

1. Introducción

Una gran cantidad de problemas concurrentes pueden ser resueltos usando los conceptos de espera condicional y exclusión mutua. Por ejemplo en [2] se presenta un método relativamente general para resolver problemas de sincronización usando el *await condicional* $\langle \textbf{await } B \rightarrow S \rangle$ el cual realiza ambas ideas (espera condicional y exclusión mutua). Esta construcción, sin embargo, no se espera que sea parte de la implementación final del programa concurrente, dado que implementar este tipo de instrucción *de manera general* es prohibitivamente ineficiente.

Diversos mecanismos de sincronización se han propuesto los cuales son menos abstractos que el await condicional y que pueden usarse para implementar esta construcción. Esta perdida de abstracción introduce nuevas necesidades de prueba de los programas, incrementando la complejidad de estos y por consiguiente generando muchas nuevas posibilidades de cometer errores. La presencia de errores sutiles, tanto de corrección (safety) como de progreso (en particular la posibilidad de deadlocks o livelocks) ha sido desafortunadamente más la regla que la excepción. Toda propuesta de nuevas primitivas de sincronización ha sido siempre una solución de compromiso entre facilidad de uso y posibilidad de reducir las penalidades en eficiencia.

Las *regiones críticas condicionales* (ver por ej. [1, 16]) son la construcción más cercana a los await condicionales y por lo tanto las más fáciles de usar correctamente. La única diferencia con los await condicionales es que es necesario declarar las variables compartidas sobre las cuales se operará en exclusión mutua y que podrán aparecer en las condiciones. Una construcción bastante similar en su nivel de abstracción a ésta es el *monitor con señalamiento automático* propuesto por Brinch Hansen y Hoare en [11, 12]. En este tipo de monitores el wait no se realiza sobre una variable de condición sino directamente sobre una expresión booleana. Las operaciones de signal son directamente eliminadas por lo cual la condición debe ser reevaluada cada vez que algún proceso del monitor salga de una región crítica. Hoare sugiere que usar estos monitores puede ser mucho más seguro pero presenta dudas respecto de la posible ineficiencia de reevaluar siempre la guarda. Comparando las reglas de demostración de estos monitores respecto de los distintos tipos de monitores con señalamiento explícito [6] puede corroborarse el argumento de Hoare. El trabajo de Kessels [14] plantea una manera de implementar los monitores con señalamiento automático de una manera relativamente eficiente siempre y cuando las condiciones no dependan de los parámetros de las llamadas al monitor (es decir, que sean una generalización modesta de las regiones críticas condicionales). Esta implementación es en esencia similar a la del uso de *semáforos binarios divididos* (SBS, del inglés split binary semaphores) que describiremos aquí. No sólo Kessels da esta implementación sino que sugiere también que puede hacerse aún más eficiente aplicando técnicas para eliminar chequeos innecesarios en los puntos en los cuales puede deducirse formalmente que una condición será falsa. Dado que las secciones críticas suelen ser pequeñas pero son invocadas numerosas veces, estos pequeños ahorros pueden representar incrementos drásticos en la eficiencia de los programas. El desarrollo en los últimos años de los demostradores (semi)automáticos de teoremas brinda el contexto en el cual una parte interesante de estas simplificaciones puede hacerse de manera (semi)automática, abriendo las puertas a reducir significativamente las penalidades en eficiencia por el uso de construcciones de muy alto nivel para la construcción de programas concurrentes.

Los SBS permiten implementar de manera eficiente las regiones críticas condicionales [9, 16]. La técnica SBS brinda tanto los programas que implementan las secciones críticas condicionales como los invariantes iniciales que aseguran su corrección. Aplicando la técnica a casos particulares se encuentran programas que admiten simplificaciones.

Este trabajo se concentra en el desarrollo de un procedimiento para verificar la corrección

de estas simplificaciones de forma automática. Nos enfocamos particularmente la eliminación de guardas en las sentencias condicionales finales de los programas, pero el procedimiento encontrado permite su generalización a otros tipos de simplificaciones.

El procedimiento consiste en hacer una búsqueda de nuevos invariantes que avalen la corrección de las simplificaciones. Su implementación así como una breve reseña de las herramientas usadas se hará en la sección 4.

Para hacer estas eliminaciones modelamos los programas generados por medio de sistemas de transiciones guardados. Con ello podemos aplicar al problema técnicas convencionales de generación automática de invariantes. La técnica que utilizamos es *propagación hacia atrás*[5]. Esta técnica permite constatar la invariancia de aserciones sobre los sistemas de transición. La ventaja que posee la misma sobre otras técnicas (como propagación hacia adelante) es que los invariantes obtenidos son fórmulas libres de cuantificadores. Con esto, la obtención de los mismos puede hacerse de forma mecánica con la ayuda de demostradores automáticos de teoremas para fórmulas de primer orden sin cuantificadores (o chequeadores de validez). En particular usaremos CVC Lite [3]. Este chequeador provee varias teorías interpretadas, incluyendo aritmética lineal, vectores, funciones, etc., lo cual es apropiado para la obtención de invariantes.

Las técnicas de propagación están basadas en el cálculo de un punto fijo para un transformador de fórmulas. Una de las ventajas de la propagación hacia atrás es que la secuencia de aproximaciones es usualmente finita. Desafortunadamente las fórmulas producidas en el proceso son generalmente grandes. Usaremos técnicas de simplificación implementadas en CVC Lite y en Isabelle/HOL [13] para aliviar este problema.

El artículo está estructurado como sigue. En la sección 2 describiremos la técnica SBS. En la sección 3 presentaremos las consideraciones teóricas en las cuales se basa el trabajo. En la sección 4 se describe la solución al problema en cuestión. Los resultados de aplicar esta técnica a diversos ejemplos se presentan en la sección 5. La sección 6 establece algunas conclusiones y direcciones en las cuales el trabajo puede ser continuado.

2. SBS

Los semáforos binarios pueden asegurar de manera muy simple exclusión mutua y por lo tanto son buenos candidatos para implementar regiones críticas. Una manera particular de usar los semáforos binarios provee un método para implementar regiones críticas condicionales. Describiremos aquí estas ideas brevemente, remitiendo a la literatura para una presentación más completa [1, 16, 9, 15, 4].

Un conjunto $\{s_0, \ldots, s_n\}$ de semáforos binarios se denominará *semáforo binario dividido* (SBS, del inglés *split binary semaphore*) si en cualquier momento de la ejecución del programa a lo sumo uno de ellos toma el valor 1. Esto es equivalente a requerir la invariancia de la siguiente propiedad:

$$0 \leq \langle \sum i : 0 \leq i \leq n : s_i \rangle \leq 1$$

Toda sección crítica comenzará entonces con una operación P sobre alguno de los elementos del SBS y terminará con un V sobre un elemento del mismo conjunto (no necesariamente el mismo). El invariante garantiza entonces exclusión mutua entre estas dos operaciones.

Además de la exclusión mutua, los SBS satisfacen la siguiente *regla del dominó* [15]. Si la ejecución de una sección crítica termina con una operación V sobre un semáforo s, entonces la próxima operación P deberá ocurrir sobre el mismo semáforo s. Esto permite asumir la precondición de cualquier operación V como postcondición de su correspondiente operación P.

Esta regla puede formularse en términos axiomáticos como la invariancia global de la siguiente aserción:

$$\varphi_{SBS} : \langle \forall s :: s = 0 \vee I_s \rangle$$

siempre que la aserción I_s sea válida antes de cada operación V asumiendo que cada I_s vale luego de su correspondiente operación P.

Para implementar regiones críticas condicionales usaremos los SBS de la siguiente manera. Asociaremos un semáforo binario del SBS a cada condición diferente. Será también necesario un semáforo más para el caso en el cual ninguna condición es válida. Luego, toda región crítica estará dinámicamente prefijada por un P asociado con su precondición. Debe tenerse cierto cuidado para introducir suficientes operaciones V para asegurar progreso. Ilustraremos el método con dos regiones condicionales para facilitar la notación. La extensión a un número arbitrario es inmediata y el resultado puede verse en la sección 4 (figura 3).

Se quieren ejecutar atómicamente las sentencias $\mathrm{S_o}$, $\mathrm{S_1}$ asumiendo como precondiciones respectivamente B_0 y B_1. Supongamos además que las regiones críticas deben preservar cierto invariante I. Usaremos entonces un SBS compuesto por dos semáforos s_0, s_1 uno para cada condición y otro m para cuando ninguna de las dos valga. Dos contadores b_0, b_1 serán necesarios para contar la cantidad de procesos comprometidos respectivamente con la ejecución de $\mathrm{P}.s_0, \mathrm{P}.s_1$ y poder así asegurar la ausencia de deadlocks.

El siguiente invariante caracteriza la solución basada en SBS

$$\begin{aligned} \varphi_{SBS} : \ & (s_0 = 0 \vee (B_0 \wedge 0 < b_0 \wedge I)) \wedge \\ & (s_1 = 0 \vee (B_1 \wedge 0 < b_1 \wedge I)) \wedge \\ & (m = 0 \vee ((\neg B_0 \vee 0 = b_0) \wedge (\neg B_1 \vee 0 = b_1) \wedge I)) \end{aligned}$$

La figura 1 muestra el programa obtenido por la técnica SBS, junto con su anotación completa, para la primer región crítica condicional.

$\underline{\mathrm{SCC_o}}$

$\mathrm{P}.m$;
$\{I \wedge (\neg B_0 \vee b_0 = 0) \wedge (\neg B_1 \vee b_1 = 0)\}$
$\underline{\text{if}}$ $B_0 \rightarrow \{I \wedge B_0 \wedge b_0 = 0 \wedge (\neg B_1 \vee b_1 = 0)\}$
skip
$\square\ \neg B_0 \rightarrow \{I \wedge \neg B_0 \wedge (\neg B_1 \vee b_1 = 0)\}\ b_0 := b_0 + 1$;
$\{I \wedge (\neg B_0 \vee b_0 = 0) \wedge (\neg B_1 \vee b_1 = 0)\}$
$\mathrm{V}.m$; $\mathrm{P}.s_0$;
$\{I \wedge B_0 \wedge b_0 > 0\}\ b_0 := b_0 - 1$
$\underline{\text{fi}}$;
$\{I \wedge B_0\}\ \mathrm{S_o}\ \{I\}$
$\underline{\text{if}}$ $B_0 \wedge b_0 > 0 \rightarrow \{I \wedge B_0 \wedge b_0 > 0\}\ \mathrm{V}.s_0$
$\square$ $B_1 \wedge b_1 > 0 \rightarrow \{I \wedge B_1 \wedge b_1 > 0\}\ \mathrm{V}.s_1$
$\square$ $(\neg B_0 \vee b_0 = 0) \wedge (\neg B_1 \vee b_1 = 0)$
$\rightarrow \{I \wedge (\neg B_0 \vee b_0 = 0) \wedge (\neg B_1 \vee b_1 = 0)\}\ \mathrm{V}.m$
$\underline{\text{fi}}$

Figura 1: Componente generada por la técnica SBS

3. Verificación de Invariantes

Las siguientes definiciones y conceptos están basados en los trabajos [17, 5]. Sea Σ un lenguaje de primer orden con símbolos interpretados en dominios concretos como booleanos, enteros, reales, etc. Sea $\mathcal{F}$ el conjunto de fórmulas de primer orden sobre Σ con variables libres contenidas en un conjunto finito de símbolos de variables $\mathcal{V}$ (tipadas). Usaremos como abreviatura de la secuencia de variables $x_0, \cdots, x_n$ a $\bar{x}$. De la misma forma, si un término del lenguaje Σ (posiblemente una formula en $\mathcal{F}$) posee variables libres en $\{x_0, \cdots, x_n\} \subseteq \mathcal{V}$, lo escribiremos como $t(\bar{x})$.

Un *sistema de transiciones guardado* es una tupla $\mathcal{S} = \langle \mathcal{V}, \Theta, \mathcal{L}, \mathcal{T} \rangle$, donde $\mathcal{V} = \{x_0, \cdots, x_n, vc\}$ (vc se denomina variable de control o "program counter"), $\Theta \in \mathcal{F}$ es la condición inicial del sistema, $\mathcal{L}$ es un conjunto finito de valores denominados locaciones y $\mathcal{T}$ es un conjunto finito de transiciones. Cada $\tau \in \mathcal{T}$ puede especificarse como

$$vc = l_\tau \wedge \gamma_\tau(\bar{x}) \;\longmapsto\; \bar{x} := \bar{e}_\tau(\bar{x})\,; vc := l'_\tau \tag{1}$$

con $l_\tau \in \mathcal{L}$ el origen de la transición, $l'_\tau \in \mathcal{L}$ el destino de la misma, $\gamma_\tau(\bar{x})$ una fórmula con variables libres contenidas en $\mathcal{V}/\{vc\}$ denominada *guarda de la transición* y $\bar{e}_\tau(\bar{x})$ es una secuencia de términos de Σ con variables libres contenidas en $\mathcal{V}/\{vc\}$ de la misma longitud que $\bar{x}$. Para cada transición, la fórmula $\gamma_\tau(\bar{x})$ denota la condición necesaria para la ejecución de la misma y $\bar{x} := \bar{e}_\tau(\bar{x})$ es una asignación múltiple que denota la transformación del estado producida por dicha ejecución.

Dado un sistema $\mathcal{S} = \langle \mathcal{V}, \Theta, \mathcal{L}, \mathcal{T} \rangle$ definimos como la *precondición mas débil de una transición*[1] al transformador de fórmulas

$$\mathrm{wp}\,(\tau, \varphi)(\bar{x}) = \gamma_\tau(\bar{x}) \rightarrow \varphi(\bar{e}_\tau(\bar{x}))$$

con $\tau \in \mathcal{T}$ y φ una fórmula con variables libres contenidas en $\mathcal{V}/\{vc\}$.

Como el conjunto de locaciones $\mathcal{L}$ es finito, una fórmula $\phi \in \mathcal{F}$ puede escribirse como

$$\phi(\bar{x}, vc) = \bigwedge_{l \in \mathcal{L}} vc = l \rightarrow \phi(\bar{x}, l) \;\; . \tag{2}$$

Para un sistema $\mathcal{S}$, la fórmula $\phi(\bar{x}, vc)$ denotará un conjunto de estados para cada locación en $\mathcal{L}$ y cada formula $\phi(\bar{x}, l)$ denotará el conjunto para la locación l en particular. Escribiremos la fórmula $\phi(\bar{x}, l)$ como $\phi_l(\bar{x})$. Utilizando esta notación definiremos el siguiente transformador de fórmulas:

$$\mathrm{WP}\,(\mathcal{T}, \phi)(\bar{x}, vc) = \bigwedge_{l \in \mathcal{L}} vc = l \;\rightarrow \bigwedge_{\substack{\tau \in \mathcal{T} \wedge \\ l_\tau = l}} \mathrm{wp}\,(\tau, \phi_{l'_\tau})(\bar{x}) \;\; . \tag{3}$$

Este transformador define la *precondición más débil de un sistema de transiciones.*

Sea $\mathfrak{R}$ una teoría sobre el lenguaje Σ. Dado un sistema $\mathcal{S} = \langle \mathcal{V}, \Theta, \mathcal{L}, \mathcal{T} \rangle$, una fórmula $\varphi \in \mathcal{F}$ es un *invariante inductivo* del sistema si $\mathfrak{R} \models \Theta \rightarrow \varphi$ y $\mathfrak{R} \models \varphi \rightarrow \mathrm{WP}\,(\mathcal{T}, \varphi)$. Ya que la teoría $\mathfrak{R}$ es fija, no la mencionaremos explícitamente cuando hablemos de validez y satisfabilidad en $\mathfrak{R}$. Una fórmula $\phi \in \mathcal{F}$ será un *invariante* (a secas) si existe un invariante inductivo φ tal que $\models \varphi \rightarrow \phi$.

Para un transformador de fórmulas monótono $\Gamma : \mathcal{F} \mapsto \mathcal{F}$, escribiremos el mayor punto fijo como $\nu X.\Gamma(X)$. Su significado será el usual [10, 5].

[1] Esta es la definición usual de "weakest precondition" de la asignación $\bar{x} := \bar{e}_\tau(\bar{x})$ si se cumple γ_τ.

Dada una fórmula $\phi \in \mathcal{F}$ definiremos el transformador de fórmulas monótono

$$\mathcal{B}(Y) \triangleq \phi \wedge \mathrm{WP}\,(\mathcal{T}, Y)\ . \tag{4}$$

Sea $\varphi_{\mathcal{B}} = \nu X.\mathcal{B}(X)$ el mayor punto fijo de este transformador en un sistema $\mathcal{S}$ dado. Entonces $\varphi_{\mathcal{B}}$ es el predicado más débil tal que $\models \varphi_{\mathcal{B}} \rightarrow \phi$ y $\models \varphi_{\mathcal{B}} \rightarrow \mathrm{WP}\,(\mathcal{T}, \varphi_{\mathcal{B}})$. Por lo tanto, si $\models \Theta \rightarrow \varphi_{\mathcal{B}}$ entonces ϕ es un invariante y $\varphi_{\mathcal{B}}$ es un invariante inductivo del sistema.

Al ser $\mathcal{B}$ un transformador monótono, si la secuencia $\varphi^{(0)}, \varphi^{(1)}, \cdots, \varphi^{(i+1)}, \cdots$

$$\underbrace{\mathit{True}}_{\varphi^{(0)}} \leftarrow \underbrace{\mathcal{B}(\varphi^{(0)})}_{\varphi^{(1)}} \leftarrow \cdots \leftarrow \underbrace{\mathcal{B}(\varphi^{(i)})}_{\varphi^{(i+1)}} \cdots \tag{5}$$

converge en una cantidad finita de pasos (comenzando de *True*), entonces su límite es $\varphi_{\mathcal{B}}$. Con esta propiedad es posible explorar el espacio abstracto de estados utilizando la técnica de *propagación hacia atrás* [5]: dada una fórmula ϕ (denominada invariante candidato) obtendremos el punto fijo $\varphi_{\mathcal{B}}$ si la secuencia converge en una cantidad finita de pasos. Entonces, para verificar que ϕ es un invariante del sistema, tendremos que demostrar $\models \Theta \rightarrow \varphi_{\mathcal{B}}$. Además, si algún $\varphi^{(i)}$ de la secuencia calculada verifica $\not\models \Theta \rightarrow \varphi^{(i)}$ (la fórmula $\Theta \rightarrow \varphi^{(i)}$ es no satisfacible) entonces, por monoticidad de la secuencia, el punto fijo $\varphi_{\mathcal{B}}$ también lo verificará. Si esto sucede podemos concluir que la fórmula ϕ no es un invariante del sistema.

4. Proceso de Eliminación de Guardas

Las implementaciones de regiones críticas condicionales que brinda la técnica SBS permiten en general ciertas optimizaciones de los programas resultantes. En la mayor parte de los programas generados para distintos problemas se verifica que algunas condiciones en el comando guardado final nunca serán satisfechas al ser evaluadas en todos los posibles estados de ejecución [9, 16]. Este trabajo se concentra en la eliminación de estas guardas superfluas dentro de los programas generados.

En general, para decidir que condiciones pueden ser eliminadas, modelamos la ejecución de los programas generados con sistemas de transiciones guardados. Resumiendo, el proceso de eliminación de guardas consiste de las siguientes etapas:

1. A partir de una especificación de un problema particular de regiones críticas condicionales generamos un sistema de transiciones que modela la ejecución de los programas obtenidos mediante la técnica SBS, y un invariante inductivo sobre el mismo que asegura la exclusión mutua condicional.

2. Eligiendo una condición (del comando guardado final) y fortaleciendo el invariante obtenemos un invariante candidato que captura, no solo la exclusión mutua condicional, sino también la imposibilidad de ejecución de la condición a eliminar.

3. Finalmente, utilizando la técnica de propagación hacia atrás, verificamos si el invariante candidato es un invariante del sistema. Con ello podemos decidir si la guarda del programa se puede eliminar.

Las etapas 1 y 2 fueron implementadas en el módulo *Generador de Transiciones* escrito en lenguaje ML. La etapa restante fue implementada en el modulo *Verificador de Invariantes* escrito en el mismo lenguaje y utilizando como proceso externo el probador de teoremas

CVC Lite [3]. Para hacer más eficiente el cálculo del punto fijo (ecuación 5 en página anterior) realizado por este módulo, incluimos técnicas de simplificación del sistema transiciones y de las formulas obtenidas en cada paso de este cálculo. Las primeras fueron implementadas en ML (dentro del primer módulo) y las segundas utilizando CVC Lite y el probador de teoremas Isabelle [13] junto con algunas técnicas de simplificación propias codificadas en ML (dentro del segundo módulo). La figura 2 muestra un diagrama esquemático de todo el proceso.

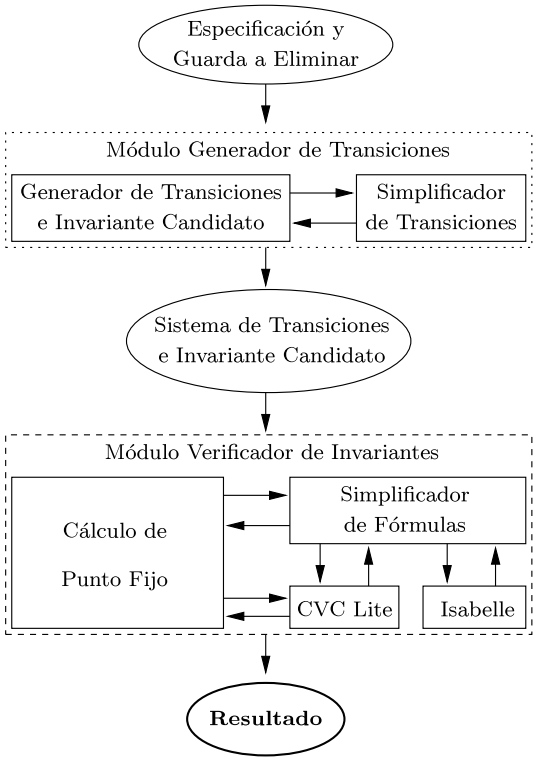


Figura 2: Diagrama del Proceso.

A continuación describiremos los detalles de implementación de ambos módulos.

Generador de Transiciones Este módulo genera el sistema de transiciones y el invariante candidato a partir de una especificación de un problema de regiones críticas condicionales y la información del comando guardado final que se desea eliminar. En su forma más general, un problema de regiones críticas condicionales puede ser especificado como:

- una secuencia de m programas $S_0, \cdots, S_{m-1}$ a ejecutarse en exclusión mutua condicional,
- m condiciones $B_0, \cdots, B_{m-1}$ de ejecución para los programas anteriores respectivamente,
- un invariante global del sistema I que debe ser mantenido por las ejecuciones de dichos programas[2].

Para simplificar la implementación del generador de invariantes, vamos a pedir que los programas $S_0, \cdots, S_{m-1}$ sean simples asignaciones múltiples.

[2]Las condiciones y el invariante global son redundantes, en el sentido que unas pueden ser obtenidos del otro de forma mecánica o viceversa. Esta previsto agregar esta funcionalidad en versiones futuras del software.

A partir de estos datos iniciales, la técnica SBS brinda m nuevos programas $SCC_0, \cdots$, SCC_{m-1} tal que, si son ejecutados de forma concurrente, verificarán la propiedad de exclusión mutua y el invariante global. La figura 3 muestra el esquema de un programa SCC_i generado por SBS.

$\underline{SCC_i}$

P.s_m ;
if $B_i \rightarrow$ skip
$\square\ \neg B_i \rightarrow b_i := b_i + 1$;
V.s_m ;
P.s_i ;
$b_i := b_i - 1$
fi;
S_i ;
if $B_0 \wedge b_0 > 0 \rightarrow$ V.s_0
$\vdots$ $\vdots$
$\square\ B_{m-1} \wedge b_{m-1} > 0 \rightarrow$ V.s_{m-1}
$\square\ \bigwedge_{0 \leq j < m} \neg B_j \vee b_j = 0 \rightarrow$ V.s_m
fi

Figura 3: Programa generado por SBS.

Analizando estos programas puede observarse que cualquier proceso comienza una traza de ejecución con una operación P y la finaliza en una operación V. Esto es una característica de la técnica SBS: semáforos binarios divididos asegura exclusión mutua entre pares de sentencia P y V, esto es, como máximo un semáforo está activado en cualquier punto de ejecución [9]. Además, por la *regla del dominó* [15] podemos asegurar que dinámicamente cada operación V será seguida por su correspondiente operación P (ejecutada por otro proceso) aplicada sobre el mismo semáforo. En base a estos hechos, modelamos cada traza de ejecución posible sobre cada programa SCC_i como una transición del sistema a generar (con la forma de la ecuación 1) y cada locación identificará que semáforo está activo ($\mathcal{L} = \{s_0, \cdots, s_m\}$).

Para obtener las guardas γ_τ de cada transición, encontramos la condición que se debe satisfacer al ejecutar la traza correspondiente. Estas condiciones vienen dadas por las guardas en los comandos guardados inicial y final que se evalúan al recorrer las trazas. Como, la interpretación de transiciones requiere que la guarda de cada una sea evaluada con anterioridad, para obtenerlas propagaremos hacia atrás las condiciones de los comandos guardados con el clásico transformador *weakest precondition* [10]. Por ejemplo, en el anterior programa SCC_i, la transición desde P.s_m hasta V.s_0 tendrá la guarda $\gamma_\tau : B_i \wedge \mathrm{wp}\,(S_i, B_0 \wedge b_0 > 0)$.

Las asignaciones de cada transición pueden obtenerse directamente calculando el cambio de estado producido por la traza correspondiente. Por ejemplo, en el anterior programa SCC_i, la transición desde P.s_i hasta V.s_0 tendrá las asignaciones $b_i := b_i - 1; S_i; vc := s_0$. En este caso, la transición total resultante será $vc = s_i \wedge \mathrm{wp}\,(b_i := b_i - 1\,; S_i, B_0 \wedge b_0 > 0) \longmapsto b_i := b_i - 1\,; S_i\,;$ $vc := s_0$.

Existen problemas de exclusión mutua condicional que requieren agregar además en su especificación ciertas *precondiciones de contorno* para algunos programas de la secuencia $S_0, \cdots, S_{m-1}$. Esto sucede por ejemplo en el caso del clásico problema "Lectores/Escritores" [1] donde los programas correspondientes a las salidas de los lectores y escritores tienen como precondición la existencia de al menos un proceso del mismo tipo ejecutándose. En la especificación del problema se pueden agregar estas precondiciones de contorno como $r > 0$ y $w > 0$ respectivamente.

La manera en que incluimos las precondiciones de contorno en el sistema de transición generado es fortaleciendo las guardas de las transiciones de los programas que las contienen: si el programas S_i tiene una precondición de contorno P, tomamos todas las transiciones generadas a partir de este programa y reemplazamos cada guarda γ_τ por $P \wedge \gamma_\tau$ en dichas transiciones.

La generación del invariante candidato se produce en dos pasos: primero se genera el invariante inductivo que asegura la exclusión mutua condicional y después se fortalece el mismo de forma que capture la imposibilidad de ejecución de la guarda a eliminar.

Para implementar el primer paso utilizamos el mismo invariante que brinda la técnica SBS: debido a que asociamos semáforos con locaciones, este invariante sirve perfectamente para el sistema de transiciones generado y además puede demostrarse que es inductivo. Dada una especificación de un problema de regiones críticas condicionales dicha técnica genera el invariante inductivo:

$$\varphi_{SBS} : \bigwedge_{0 \leq i \leq m} vc = s_i \rightarrow \varphi_{s_i} \tag{6}$$

con

$$\varphi_{s_i} : \; B_i \wedge b_i > 0 \wedge I \quad \text{con } 0 \leq i < m \;, \qquad \varphi_{s_m} : \; (\bigwedge_{0 \leq j < m} \neg B_j \vee b_j = 0) \wedge I$$

e I el invariante global de la especificación.

Para obtener el invariante candidato fortalecemos φ_{SBS} con la precondición más débil (usando el transformador wp) de la guarda a eliminar negada, sobre la trazas que llegan a dicha guarda. Con esto, el invariante candidato resultante denotará la imposibilidad de ejecución de las trazas involucradas. Por ejemplo, si queremos eliminar la guarda $\Box B_k \wedge b_k > 0$ (para algún $k \in \{0, \cdots, m-1\}$) en el programa S_i, el invariante candidato ϕ será:

$$\phi : \; \varphi_{SBD} \wedge F_m \wedge F_i$$

con

$$F_m : \; vc = s_m \wedge B_i \; \rightarrow \; wp.\, S_i.\, (\neg B_k \vee b_k = 0)$$
$$F_i : \; vc = s_i \; \rightarrow \; wp.\, (b_i := b_i - 1; S_i).\, (\neg B_k \vee b_k = 0)$$

siendo F_m el fortalecimiento correspondiente a la traza que comienza en el comando P.s_m y F_i el correspondiente a la traza que comienza en P.s_i, ambas terminando en la guarda a eliminar.

Además, el módulo efectúa simplificaciones en el conjunto de transiciones resultantes. Por ejemplo, una de estas consiste en la eliminación de la transición envuelta en la guarda a eliminar. Para una descripción detallada de las mismas y sus demostraciones de corrección remitirse a [4].

Cabe aclarar que, en la implementación del módulo, las locaciones se representaron con números enteros ($\mathcal{L} = \{0, \cdots, m\}$) y usando el resultado en la ecuación 2 (página 5) las formulas se almacenaron en arreglos indexados en $\mathcal{L}$: en general una fórmula $\phi(\bar{x}, vc) \in \mathcal{F}$ será representada por $[\phi_0(\bar{x}), \cdots, \phi_m(\bar{x})]$ y para el caso particular del invariante φ_{SBS} su representación será $[\varphi_{s_0}, .., \varphi_{s_m}]$. Este diseño estuvo inspirado en el trabajo [17].

Verificador de Invariantes Este módulo toma el sistema de transiciones generado junto con el invariante candidato más una condición inicial inicial Θ y verifica el invariante.

Dadas las fórmulas ϕ y φ (ambas representadas como arreglos) implementamos el operador $\mathcal{B}$ (ecuación 4) como muestra la figura 4 (parte izquierda).

En cada iteración la función calcula el conjunto $\{\tau \in \mathcal{T} : l_\tau = i\}$ (conjunto de transiciones que parten de i) y el resultado es obtenido utilizando la ecuación 3. La función $\mathfrak{R}$-simplify hace simplificaciones en las fórmulas y siempre devuelve fórmulas equivalentes. Esta función

function $\mathcal{B}(\phi, \mathcal{T}, \varphi)$
  for $i \in \mathcal{L}$ **do**
    $\mathcal{T}_i := \{\tau \in \mathcal{T} : l_\tau = i\}$;
    $\varphi'[i] := \phi[i] \ \wedge \ \bigwedge_{\tau \in \mathcal{T}_i} \mathrm{wp}\,(\tau, \varphi[l'_\tau])$;
    $\varphi'[i] := \mathfrak{R}\text{-simplify}(\varphi'[i])$;
  end for
  return φ' ;

function backPropagation$(\phi, \mathcal{T}, k)$
  $\varphi := [\mathit{True}, \cdots, \mathit{True}]$;
  $i := 0$;
  while $i < k$ **do**
    if $\not\models \Theta \rightarrow \mathcal{B}(\phi, \mathcal{T}, \varphi)$ **then**
      return unsat$(\mathcal{B}(\phi, \mathcal{T}, \varphi))$;
    else if $\models \varphi \rightarrow \mathcal{B}(\phi, \mathcal{T}, \varphi)$ **then**
      return converge(φ) ;
    else
      $\varphi := \mathcal{B}(\phi, \mathcal{T}, \varphi)$;
      $i := i + 1$;
    end if
  end while
  return noconverge(φ) ;

Figura 4: Programas del Verificador de Invariantes

está implementada utilizando CVC Lite y técnicas de simplificación propias. Para más detalles referirse a [4].

La función backPropagation (figura 4 parte derecha) realiza el cálculo de punto fijo utilizando el operador $\mathcal{B}$. La misma consta básicamente de un bucle donde se calculan las fórmulas $\varphi^{(i)}$ de la ecuación 5. La variable φ almacena estos valores. En cada paso de iteración, si la fórmula $\Theta \rightarrow \mathcal{B}(\phi, \mathcal{T}, \varphi)$ no es satisfacible (chequeando $\not\models \Theta \rightarrow \mathcal{B}(\phi, \mathcal{T}, \varphi)$ con CVC Lite) la función devuelve el valor unsat$(\mathcal{B}(\phi, \mathcal{T}, \varphi))$ indicando que el candidato ϕ no es invariante (ver sección 3). Si esto no sucede, el programa verifica si se ha llegado al punto fijo chequeando $\models \varphi \rightarrow \mathcal{B}(\phi, \mathcal{T}, \varphi)$ (también con CVC Lite). Si esto sucede se devuelve el valor converge(φ) el cual almacena dicho punto fijo. En otro caso se almacena el resultado del operador $\mathcal{B}$. El programa ejecuta a lo sumo k iteraciones, devolviendo el valor noconverge(φ) en el caso que se alcance este límite, indicando que no se pudo decidir la invariancia de ϕ.

5. Resultados

El procedimiento desarrollado fue aplicado a diversos ejemplos clásicos de la programación concurrente. Los ejemplos considerados fueron Semáforos Generales (implementación de semáforos generales con binarios), Productor/Consumidor (Productor/Consumidor que intercambian elementos por medio de un buffer acotado), Productor/Consumido Goloso (ídem pero consumiendo de a 3 elementos), Productor/Consumido Goloso M (ídem pero consumiendo de a m elementos con m una constante) y Lectores/Escritores. La formulación de estos problemas puede encontrarse en [1, 16, 9, 4]. El software más las especificaciones de entrada para los ejemplos pueden encontrarse en `http://www.cs.famaf.unc.edu.ar/~damian/publications/sbdinv/programs/programs.tgz`.

El procedimiento encontró todas las guardas que se podían eliminar (los resultados son los mismos que los obtenidos en la literatura) y en tiempos breves (algunos segundos) para todos los casos, excepto para la eliminación de una guarda en el problema de Productor/Consumidor consumiendo de a m elementos. Notar que este problema es una generalización de los dos anteriores, en los cuales el método terminó exitosamente. Haciendo el cálculo del punto fijo a mano y con la ayuda del mismo software (el software permite ver las formulas $\varphi^{(i)}$ en cada iteración), se puede deducir que las fórmulas de la ecuación 5 claramente no convergen en un

número finito de pasos.

Los resultados fueron obtenidos utilizando una PC con procesador Athlon XP de 1400 MHz. y 256 MB de memoria RAM. El detalle de los mismos (guardas que se pudieron eliminar, run time, cantidad de iteraciones, etc.) puede encontrarse en [4].

6. Conclusiones y Trabajos Futuros

Las regiones críticas condicionales son un patrón de alto nivel para la programación concurrente. La mayor parte de los lenguajes de programación no provee este tipo de construcciones como primitivas debido a que son difíciles de implementar de forma eficiente. Este trabajo muestra un método para optimizar automáticamente implementaciones de regiones críticas condicionales realizadas mediante la técnica SBS. Dado que muchos problemas concurrentes pueden ser resueltos con estas construcciones, el método puede ser muy útil a la hora de crear compiladores que implementen regiones críticas condicionales.

En este trabajo se probó la eficiencia del procedimiento sobre algunos problemas clásicos de concurrencia. Si bien no se dispone de una caracterización de los problemas para los cuales el algoritmo de propagación converge, este funciona bien para una gran cantidad de ejemplos. En su estado actual, el algoritmo podría ser aplicado para la optimización de programas simplemente limitando el número de pasos usando alguna constante que podría determinarse empíricamente. Si bien esto puede dejar fuera programas convergentes, este fenómeno no necesariamente puede ser resuelto en general, dado que el algoritmo tiene que chequear también implicaciones en la aritmética, lo cual no siempre es decidible.

El algoritmo puede mejorarse usando widenings e interpretación abstracta [17, 5, 7] para los casos en los cuales no converge. Hasta el momento solo se intentó utilizar interpretación abstracta sobre el dominio de los poliedros convexos [8]. Con este método solo se encontraron invariantes más débiles que los obtenidos mediante la técnica SBS (ecuación 6) lo cual no permite realizar nuevas simplificaciones. Quedan pendientes para futuros trabajos el empleo de otros dominios de interpretación abstracta.

Referencias

[1] G. Andrews. *Foundations of Multithreaded, Parallel, and Distributed Programming.* Addison-Wesley, Reading, Massachusetts, USA, 1999.

[2] G. R. Andrews. A method for solving synchronization problems. *Sci. Comput. Program.*, 13(1):1–21, 1989.

[3] C. Barrett and S. Berezin. CVC Lite: A new implementation of the cooperating validity checker. In R. Alur and D. A. Peled, editors, *Proceedings of the* 16^{th} *International Conference on Computer Aided Verification (CAV '04)*, volume 3114 of *Lecture Notes in Computer Science*, pages 515–518. Springer-Verlag, July 2004. Boston, Massachusetts.

[4] D. Barsotti and J. O. Blanco. (Im)proving split binary semaphores. Tecnical Report. Available at `http://www.cs.famaf.unc.edu.ar/~damian/publicaciones/sbdinv/SBDwip_ext.pdf`, 2007.

[5] N. Bjorner, A. Browne, and Z. Manna. Automatic generation of invariants and intermediate assertions. *Theor. Comput. Sci.*, 173(1):49–87, 1997.

[6] P. A. Buhr, M. Fortier, and M. H. Coffin. Monitor classification. *ACM Comput. Surv.*, 27(1):63–107, 1995.

[7] P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*, pages 238–252, 1977.

[8] P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among variables of a program. In *Conference Record of the Fifth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 84–97, Tucson, Arizona, 1978. ACM Press, New York, NY.

[9] E. W. Dijkstra. A tutorial on the split binary semaphore. `http://www.cs.utexas.edu/users/EWD/ewd07xx/EWD703.PDF`, Mar. 1979.

[10] E. W. Dijkstra and C. S. Scholten. *Predicate calculus and program semantics.* Springer-Verlag New York, Inc., 1990.

[11] P. B. Hansen. A programming methodology for operating system design. In *IFIP Congress*, pages 394–397, 1974.

[12] C. A. R. Hoare. Monitors: An operating system structuring concept. *Commun. ACM*, 17(10):549–557, 1974.

[13] Isabelle Theorem Prover home page. `http://isabelle.in.tum.de`, 2007.

[14] J. L. W. Kessels. An alternative to event queues for synchronization in monitors. *Commun. ACM*, 20(7):500–503, 1977.

[15] A. Martin and J. van de Snepscheut. Design of synchronization algorithms. *Constructive Methods in Computing Science*, pages 445–478, 1989.

[16] F. B. Schneider. *On Concurrent Programming.* Graduate texts in computer science. Springer-Verlag New York, Inc., 1997.

[17] A. Tiwari, H. Rueß, H. Saïdi, and N. Shankar. A technique for invariant generation. In T. Margaria and W. Yi, editors, *TACAS 2001 - Tools and Algorithms for the Construction and Analysis of Systems*, volume 2031, pages 113–127, Genova, Italy, Apr. 2001. Springer-Verlag.

Induction Trees for

Automatic Word Classification

Daniela López De Luise
AI Group , Facultad de Ingeniería, Universidad de Palermo (UP)
Ciudad Autónoma de Buenos Aires – Argentina
lopezdeluise@yahoo.com.ar

and

Juan M. Ale
Facultad de Ingeniería, Universidad de Buenos Aires(UBA)
Ciudad Autónoma de Buenos Aires – Argentina
ale@acm.org.ar

Abstract

This work studies induction tree application for certain word category detection by simple morpho-syntactical descriptors that are proposed here. The classification power for these new descriptors with and without stemming is also studied. Finally, results show that classification prediction power is good when stem is coordinated with a short list of descriptors.

Keywords: machine learning, lexical categorization, morphology, syntax

Resumen

En este trabajo estudia el uso de árboles de inducción para la detección de ciertos tipos de palabras usando algunos descriptores morfosintáctico propuestos. También se estudia el poder de clasificación de estos nuevos descriptores con y sin extracción de raíces de palabras (stemming). Finalmente, se muestra en los resultados que el poder de predicción de la clasificación es bueno cuando se combinan stemming con algunos de los descriptores presentados.

Palabras claves: aprendizaje automático, clasificación de palabras, mofología, sintaxis

1. INTRODUCTION

It is hard to perform an efficient handling of digital documentation due to several phenomena as synonymy (different words with similar meaning), polysemy (a word with two or more meanings), anaphoras (implicit mentions by means of demonstrative pronouns), metaphors (use of a word with a meaning or in a context different from the habitual one), metonymy (rhetorical figure that consists of transferring the meaning of a word or phrase to another word or phrase with different meaning, with semantic or logical proximity) [10], misspellings, punctuation, neologisms, foreigner words and differences between linguistic competence (based in grammar rules) and actuation (the way grammar is used by a native speaker) [2]. Many approaches have been used to solve these problems, some of them are:
-Exhaustive tables of words or punctuation, optionally combined with lexical knowledge databases such as WordNet (to process using synonyms) [10].
-Exhaustive text revision to extract and classify errors in texts [2].
-Use of a corpus of traditionally detectable mistakes in the language [2].
-Normative [2].
-Style books [2].
-Scoring synonymy degree of expressions [4].
-Contextual information processing [13].

Based on those strategies, several applications and studies have been performed: for correcting documents [14], classification of documents, written text analysis, inflectional language[1] analysis [17], statistical machine translation [12], text summarization [10], automatic grammar and style checking [2] automatic translation [4], etc., even covering areas like statistical modeling of speech [8]. To perform such activities it is very useful to be able to automatically detect the word lexical category (if a word is a noun, article, verb, etc.). Sometimes this detection is part of the global approach as in the case of the text checking presented in [6], whereas in other cases are special developments as in [2], or [7], but always with complex semantic management or with long linguistic inference procedures. This paper proposes a set of morpho-syntactical descriptors for words, using just local information, to be used to automatically find out the actual lexical category of certain words with reasonable precision. The set of morpho-syntactical descriptors defined here are combined with stemming algorithmic [15] to get invariant radixes as extra descriptors. This proposal uses also an Induction Tree. Although Induction Trees[2] can be used for learning in many areas [11], they are applied here to word classification. An induction tree is a model of some basic characteristics of a dataset extracted by an induction process on instances. It is used due to its flexibility and its power to apply the acquired knowledge to new concrete instances.

Because the Web is a kind of text repository, traditional morpho-syntactical processing had to overcome new problems (specific problems for internet documentation): It will be required to adapt processing to activities such as Web Services [14], Information Retrieval, automatic extraction of knowledge from Web Documents [1], using Web as a

[1] languages, where words have usually several different morphological forms that are created by changing a suffix [17].

[2] From Mitchell [11]: "Decision Tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree. Learning trees can also be re-represented as sets of if-then rules to improve human readability. These learning methods are among the most popular of inductive inference algorithms and have been successfully applied to a broad range of tasks from learning to diagnose medical cases to learning to assess credit risk on loan applicants" .

corpus for automatic collocation[3] identification [16], etc. Therefore it is important to process automatically text as mentioned previously but considering the special features of web writers and readers. For that reason, all the text processed in this paper is extracted only from web pages.

Another point is that internet sets the same availability degree for sites in any language. So, the web pages covered here are taken from Spanish sites in any country.

The rest of this paper is organized as follows: section 2 describes the database and data collection procedure, section 3 describe field selection and induction tree model construction, and section 4 presents some conclusions and future work.

2. DATA ANALYSIS

In this section there is a short description of the processing steps (section 2.1), dataset and sample characteristics (sections 2.2 and 2.3 respectively).

2.1. Methodology

Four sets of web pages in Spanish were made regarding several topics. All of them were downloaded in text format. From the total number of 340 pages, 361217 words were extracted with a Java application. The output was saved as 15 plain text files. The text files were converted into Excel format to be able to use an Excel's form to manually fill in the field tipoPalabra (kind of word). The resulting files were processed with other java program to introduce the stemming column and afterward converted into csv format to be able to work with WEKA[4] software. After that, some preliminary statistics were performed with InfoStat[5] to detect the main dataset features and the csv files were processed with WEKA Explorer. An induction tree model was built from data as detailed in the following sections. Figure 1 depicts graphically all the mentioned steps.

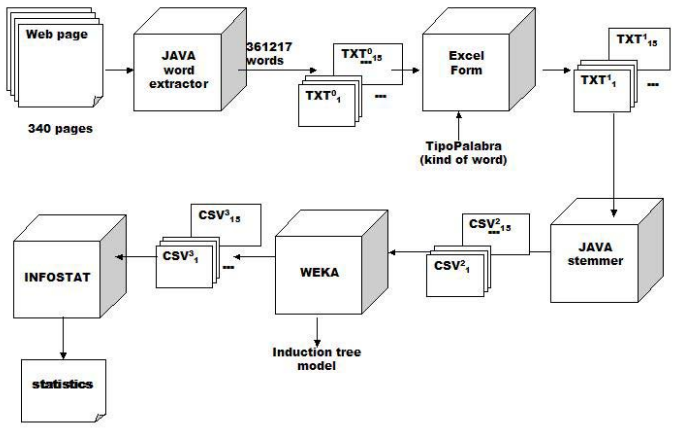


Figure 1. Flow of processing steps

[3] statistically significant word associations representing "a conventional way of saying things" [9].
[4] WEKA: open source workbench for Data Mining and Machine Learning [18].
[5] InfoStat: statistical software from a group named InfoStat in the Universidad Nacional de Córdoba.

2.2. Dataset Description

The text files were processed with a Java application. For each word, a set of 25 description fields were extracted. Therefore, each database record represent a word. The fields are detailed below:
-Continue fields: there isn't.
-Numerable fields: 10 fields were non-negative integers with a big boundary (see Table 1). All of them were discretized into fixed-size intervals, to be able to categorize and process them together with nominal fields. They were separated into 3 or 5 categories. (see Table 2).
-Discrete fields: there isn't.
-No-numeric fields: 15 fields have a domain composed by a specific set of literals (syllabus, punctuation signs, a set of predefined words or the classical binomial Yes/No). See Table 3 for details.
-Missing data: they were considered as a distinct data value and processed with the rest of the data.

Table 1. Numerable fields

Field	description
Id-caso	Web page dentifier
Web-profundidad-pagina	Number of slashes ("/") in the web page URL
Cant-ocurrencias	Times the word is repeated in the page
Cant-pal-pagina	Number of words in the page
Long-palabra	Number if characters in the word
Cant-vocales-fuertes	Number of strong vowels in the word (a, e, o)
Cant-vocales-debiles	Number ofweak vowels in the word (I, u)
Long-oracion	Number of words (in words' sentence)
Cant-numeros	Quantity of numbers (in words' sentence)
Cant-signos-especiales	Number of special characters (in words' sentence)

Table 2. Categorization

Field	categories	Max value
Web-profundidad-pagina	5	7000
Cant-ocurrencias	3	1168000
Cant-pal-pagina	5	6792600
Long-palabra	5	31000
Cant-vocales-fuertes	3	11000
Cant-vocales-debiles	3	6000
Long-oracion	5	259000
Cant-numeros	5	842000
Cant-signos-especiales	5	149000

2.3 Sample Characteristics

Data fields dependences were studied with correspondence analysis. This task was performed with InfoStat software. All the 25 fields were considered, but only a random sample of 47820 instances were processed. The independency test was performed with parameter $\alpha = 0.05$, statistic χ^2 y H_0= "independent".

Results show that:
-tipoPalabra (kind of word) is independent from tipoPag (kind of page) and siguePuntuación (punctuation follows the actual word).
-palAntTipo (kind of previous word) is independent from cantVocalesFuerte (number of strong vowels in the word).
-resaltada (the word is remarked in the text) is independent from cantVocalesFuerte (number of strong vowels).

Table 3. Results with Different Splits

Field	Description
Id-palabra	Word extracted
Tema	Main topic of the web page
Tipo-pal	One of these: sustantivo, verbo, otro (noun, verb, other respectively)
Pal-anterior	Identifier of the previous word in the sentence
Pal-ant-tipo	One of these: ninguna, otro (none, other). This indicates if it is the first word in the sentence.
Tipo-pag	One of these: indice, contenido (index, plain text).
Pais-radicacion	One of the country codes in the url: us, ar, es, cu, mx, etc.
Terminacion	Suffix of the word: ar, er, ir, or, ur, ra, re, ri, ro, ru, s, m, sa, se, si, so, su, an, en, in, on, un, cion, ciones, null.
Sigue-puntuacion	One of these: si, no (Yes, No). This value is "Si" when one the following is after the word: ".", ",", ";", ":".
Clase-pag	One of these: org, com, net, otro. Indicates the kind of domain from the URL.
Empieza-mayuscula	One of these: si, no (Yes/No). This value is "Si" when the word starts capitalized.
Resaltada	One of these: si, no (Yes/No). This value is "Si" when the word is enclosed with quotation marks or is capitalized.
Es-titulo	One of these: si, no (Yes/No). This value is "Si" when the word belongs to the first sentenece in the text.
Frase-especial	One of these: si, no (Yes/No). This value is "Si" when the sentence is enclosed with: ¡!, ¿?, <>, (), [], {} or «»
Stem	Radix of the word (according to Porter algorithm)

3. INDUCTION TREES FOR CLASSIFICATION

In this section the construction of an induction tree (using WEKA [18] software) with many parameter values is studied. The remainder of this work uses the following metrics to evaluate results [18]:

1) Metrics used for error handling evaluation

1.a) Precision: metric used in Information Retrieval (IR). It is the rate of relevant instances returned by the total of instances returned.

1.b) Recall: metric used IR. It is the rate of relevant returned by the number of relevant instances.

1.c) Recall-precision: plot used in IR with recall (x-axis) and precision (y-axis).

2) Metrics used for predictability

2.d) Kappa (κ): used to compare predictability against a random predictor. It can take from 0 to 1, being 0 the random predictor value and 1 the best predictor.

3) Metrics for confidence validation

3.e) Margin curve: a bigger margin denotes a better predictor. It is the difference between the estimated probability of the true class and that of the most likely predicted class other than the true class.

In the following, an induction tree with J4.8 algorithm is used to build a model to predict the kind of certain words based on the descriptors introduced in this paper (section 3.1), based on the descriptors and stem (section 3.2) and based only in the best descriptors and stem (section 3.3).

3.1. Classification of Words Using proposed descriptors

The J48 algorithm is used here to build the induction tree using the fields presented in 2.2. Dataset Description (except for stem field). Here, the following analysis is performed: alternate splittings of training sample, different data categorizations, influence of descriptors on the model and windowing[6].

[6] windowing is a strategy for selecting a subset of data for processing.

1) Splitting of the training sample.
Different percentages of instances were taken from the same sample to construct/validate the model by setting several splitting values. The data records were randomly extracted from the 47820 instances according to the settled percentage. The initial sampling window had 6838 instances. Results are shown in Table 4.

Table 4 Results with Different Splits

split	correctly classified	Kappa statistic
66%	70.7%	0.4582
70%	70.7%	0.4558
00%	72.7%	0.4981

It can be seen that classification improves from 66% of instances for testing (and 34% for training) to 100% for training and testing. The classification model becomes more confident.

2) Alternates for field categorization.
As part of sensitivity analysis, different categorizations for just one of the descriptor variables is performed: cantOcurrencias (number of times the word is detected within the html page). This variable is selected for this study because it is always near the tree-model root (it is important to determine the kind of word). It was evaluated with 3 and 7 bins. Results are shown in Table 5.

Table 5. Results with Different Categorizations

split: 66%

categories	7	3
correctly classified	70.9%	70.7%
incorrectly classified	29.1%	29.3%
Kappa statistic	0.4698%	0.4582%

split: 70%

categories	7	3
correctly classified	70.8%	70.7%
incorrectly classified	29.2%	29.2%
Kappa statistic	0.4716%	0.4558%

split: 00%

categories	7	3
correctly classified	74.5%	70.7%
incorrectly classified	26.5%	29.3%
Kappa statistic	0.5166%	0.4981%

The table shows the precision and total error changes due to categorization. To study the strength of this tendency, the margin-curves, precision, recall and recall-precision analysis is performed but only for nouns:

- Margin-curves for 3 and 7 categories reflect a slight tendency to join the x-axis with the instance number. It seem like each new instance makes the classifier more trustable. This tendency becomes apparent with 66% of splitting, and remains with 70% and 0% (see Figure 2).

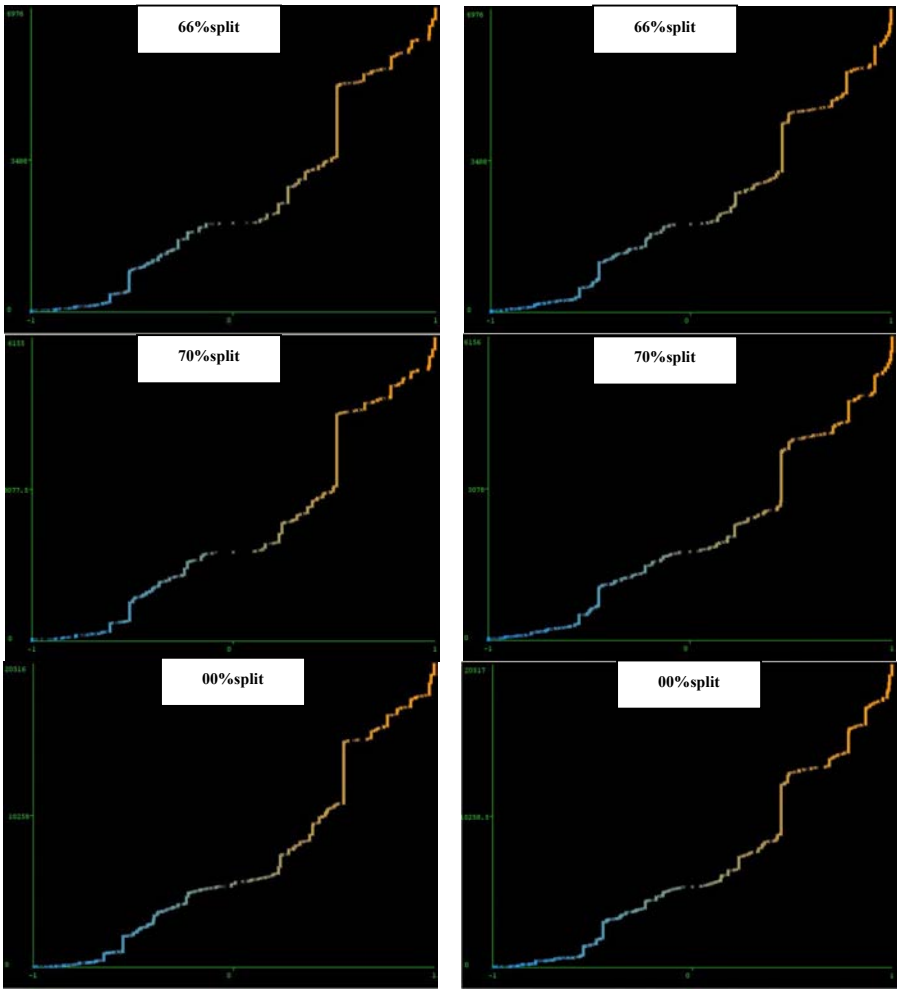
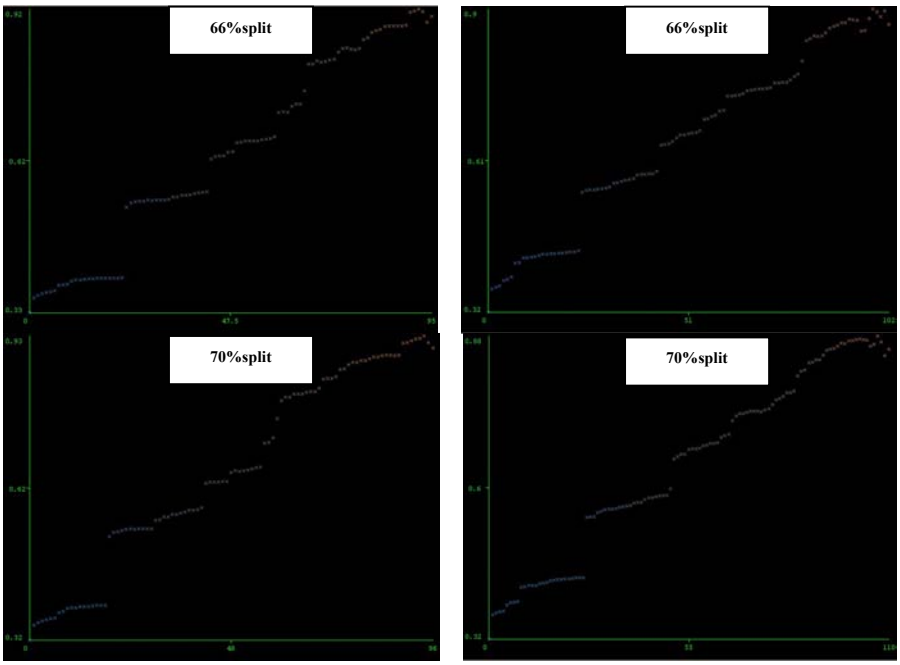


Figure 2. margin curve with 3 (on the left) and 7 categories (on the right)

- Precision-curves show that precision with 3 categories is better than with 3 categories but with 7 categories more instances are retrieved (102 against 95 with 66% of splitting). See Figure 3.



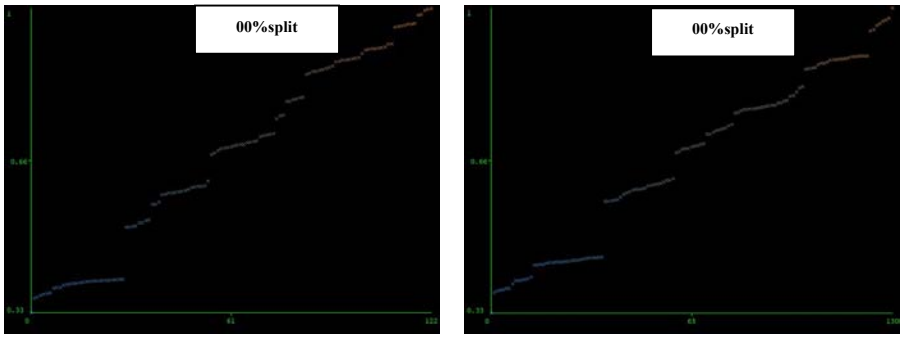


Figure 3. Precision curve for 3 (on the left) and 7 (on the right) categories

- Recall-curve presents a minimum recall value for 7 categories higher than the value for 3 categories. Conversely, the slope has a softer slope for 3 categories (see Figure 4).

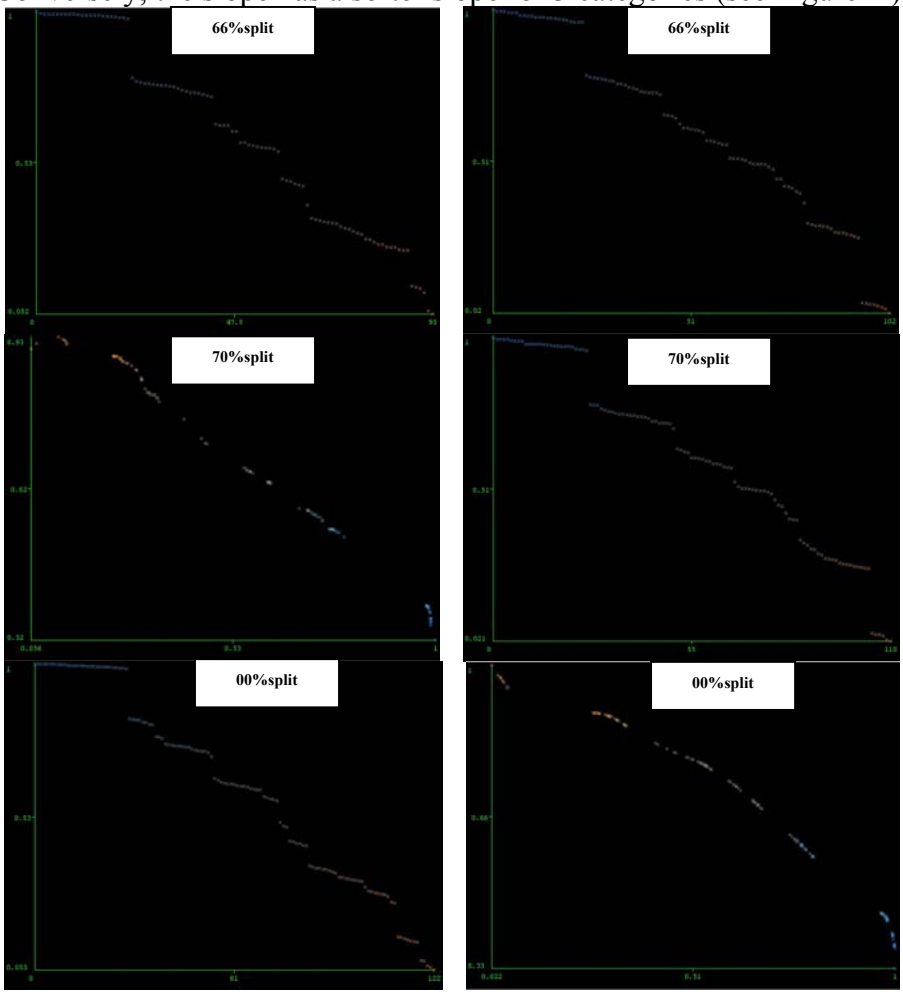


Figure 4. Recall curve for 3 (on the left) and 7 (on the right) categories

- Finally, precision-recall curve (see Figure 5), show that precision is best for 3 categories but at expense of fewer number of instances. This behavior is observed for all the splitting rates experienced (66%, 70%, 0%).

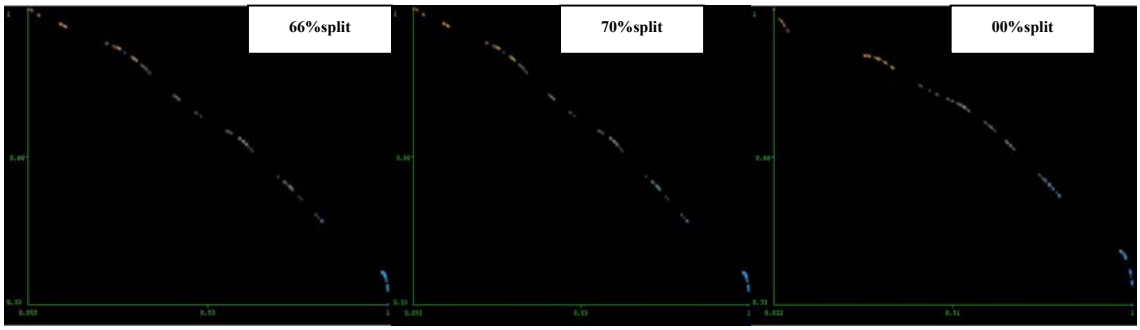


Figure 5. Precision-recall curve for 3 categories (66%, 70%, 0% split)

3)Descriptor choices

In this section alternate descriptor selection criteria are studied to find out the influence of field selection on the classification power. Table 6 shows a brief of the following analysis:

a)*Low-computational-cost fields selection*: the high-cost fields, were taken out whenever the removal did not affect the tree performance. The resulting selection was 12 descriptors. None of them involves processing all the html document. Just one of such descriptors needs sentence processing.

b)*Categorized fields selection*: nominal fields where removed and the numeric fields (categorized) were used to construct the model.

c)*Nominal fields selection*: numeric fields where removed and the nominal fields were used to construct the model.

d)*Independent fields selection*: some independent fields were taken out before constructing the model (see correspondence analysis in 2.3 Sample Characteristics). The process was repeated 5 times changing the extracted subsets according with different criteria.

Table 6. Results with Different field selections

criteria	fields	total	Classif. OK	kappa
Low cost	tema, palAntTipo, tipoPag, terminacion, empiezaMayuscula, resaltada, esTitulo, CATlongPalabra, CATcantVocalesFuerte, CATcantVocalesDebile, CATlongOracion	12	71.1%	0.4761
categorized	CATwebProfundidadPag, CATcantOcurrencias, CATcatPalPagina, CATlongPalabra, CATcantVocalesFuerte, CATcantVocalesDebile, CATlongOracion, CATcantNumeros, CATcantSignosEspeciales	9	63.82%	0.3043
nominal	tema, tipoPag, palAntTipo, paisRadicacion, terminacion, siguePuntuacion , clasePag, empiezaMayuscula, resaltada, esTitulo, fraseEspecial	12	65.63%	0.34
independent	palAntTipo, tipoPag, siguePuntuacion resaltada, CATlongPalabra, CATcantVocalesFuerte CATcantVocalesDebile, CATlongOracion	8	63.51%	0.3

As can be seen from the results in Table 6, there is a low correctly-classified rate and kappa values.

4) Instance windowing

Three windows of instances were selected. The windows were of different size and composition as described below:

a)sample 1: 47829 instances. The word-class distribution is: 6689 nouns, 2762 verbs, 11027 other class, 36 unknown class. Main characteristics of the sample: words were extracted from pages mainly with the same subtopic within the set theme. Besides, each page were longer than in the other two samples.

b)sample 2: 20515 instances. The word-class distribution is: 6392 nouns, 3050 verbs, 11054 other class, 19 unknown class. Main characteristics of the sample: pages were related to many different subtopics and typically very short in the average.

c)sample 3: 20524 instances. The word-class distribution is: 6535 nouns, 2954 verbs, 11014 other class, 21 unknown class. Main characteristics of the sample: pages were related to different subtopics but with intermediate size in the average.

The model training was performed with each sample, taking 12 data fields (4 of them categorical). Results are shown in Table 7.

Table 7. Results with Different samples

set	correctly classified	Kappa statistic
1	71.1%	0.4740
2	76.8%	0.6052
3	76.9%	0.5986

As can be seen from the table, there is a significant variation of classification power with the dataset. Those results are due the characteristic of each one. As a consequence of these characteristics, the noun rate is highest in the second sample, making the classification correctness higher than sample 1 and lower than sample 3. Kappa statistics decreases for sample 3, which has a fewer number of nouns than sample 2, even considering that sample 3 performs a bit better classification rate due to the shorter pages.

3.2. Classification Using descriptors and Stemming

The classifier behavior was studied considering stem. Sample 2 was extended with the corresponding radixes using stemming algorithm. Records with same stem were counted and those whose stem frequency is lower than 10 were eliminated from the set. The resulting set has 2316 instances.

Classification model was constructed with distinct attribute considerations: several simple global descriptors, stem and three simple descriptors, stem and six simple descriptors, stem as unique descriptor. Table 8 shows the results obtained: correctly classified rate improves with stemming combined with descriptors. Kappa value denotes that it is a better model also (κ increases up to 0.887). It can be seen that the field stem is not as good for classification by kind of word (tipoPal) as descriptors do.

Table 8. Stem with/without Descriptors

case	correctly classified	Kappa statistic
no stem	64.4%	0.4000
stem and 3 fields	90.7%	0.8500
stem and 6 fields	93.0%	0.8870
stem alone	02.5%	0.0168

3.3. Word Classification with Stemming and best Morpho-Syntactical Descriptors

The 12 best descriptors (4 of them were categorical) are selected and combined with syntactical-radixes. Such descriptors describe the topic of the document, kind of word, kind of html page, kind of previous word, word suffix, word length, number of vowels, etc.
Results with and without stemming are shown in Table 9. Here the confidence level has improved very much when considering stem.

Table 9 Descriptors with/without Stem

	with stem	without stem
correctly classified	94%	64%
Kappa statistic	0.90	0.40

4. CONCLUSIONS AND FUTURE WORK

From the previous sections some interesting conclusions can be extracted:
-Training set must have more than 20514 to get better results.
-Categorization procedure takes influence on the classifier confidence, improving it when the number of categories increases.
-The best subset of data fields have many interdependencies.
-The html-page length influences the performance. Better results are obtained with lengthy pages.
-Stem has not enough classification power by itself.
-Descriptors have not enough classification power by itself.
-A combination of stemming with detected better descriptors makes it possible to perform word classifications with good confidence levels.

Some interesting future works are:
-Repeat this analysis considering as kind of previous words: “none”, “article”, “preposition”, “pronoun”, and “other”.
-Analyze categorical field dependencies to reduce the number of variables with a kind of formula.
-Study the variations due to other field categorization criteria.
-Evaluate alternate algorithms.
-Compare results against other sources as books, magazines, etc.

REFERENCES

[1]. Alani H. et al. (2003) "Automatic Extraction of Knowledge from Web Documents", In Proc. of 2nd International Semantic Web Conference - Workshop on Human Language Technology for the Semantic Web and Web Services, Sanibel Island.

[2]. Aldezábal I. (1996) "Del analizador morfológico al etiquetador: unidades léxicas complejas y desambiguación". Procesamiento del lenguaje natural. N. 19, pp. 90-100. España.

[3]. Díaz Villa A.M. (2005) "Tipología de errores gramaticales para un corrector automático" Magazine Procesamiento del Lenguaje Natural, vol 35.

[4]. Fernández Lanza S. (2003) "Una contribución al procesamiento automático de sinonimia utilizando Prolog" PhD dissertation, Santiago de Compostela, España.

[5]. Figuerola C. G. (2000) "Categorización automática de documentos en español: algunos resultados experimentales". ReLIS, Jornadas de Bibliotecas Digitales. http://imhotep.unizar.es/jbidi/jbidi2000/14_2000.pdf

[6]. Genthial D. (1990) "Contribution of a Category Hierarchy ti the Robustness of Syntacic Parsing", 13th CoLing, vol. 2, pp. 139-144. Helsinki, Finland.

[7]. Gulla A. A. (1996) "A Sign Expansion Approach to Dynamic, Multi-purpose Lexicons", International Conference on Computational Linguistics. Proceedings of the 16th Conference on Computational Linguistics. Vol. 1. pp. 478 – 483. Copenhagen. Denmark.

[8]. Levinson S. (2006) "Statistical Modeling and Classification", AT&T Bell Laboratories, Murray Hill, New Jersey, USA. Also available at http://cslu.cse.ogi.edu/HLTSurvey/ch11node4.html.

[9]. Manning C., Schütze H. (1999) "Foundations of Statistical Natural Language Processing", Cambridge, Mass. MIT Press. ISBN 0262133601

[10]. Mateo P.L., González J.C., Villena J., Martínez J.L. (2003) "Un sistema para resumen automático de textos en castellano" DAEDALUS S.A., Madrid, España.

[11]. Mitchell T. (1997) Machine Learning, New York: WCB/Mc Graw Hill, pp. 51-80.

[12]. Nießen S., Ney H. (2000) "Improving SMT quality with morpho-syntactic analysis", in Proc. of the 18th conference on Computational linguistics – Vol. 2, pp. 1081 – 1085, Saarbrücken, Germany.

[13]. Oliveira O.N., Nunes M.G. V., Oliveira M.C. F. (1998) "Por qué no podemos hablar con una computadora?" Magazine of Sociedad Mexicana de Física., México, v. 12, pp. 1 - 9.

[14]. Platzer C., Dustdar S. (2005) "A Vector Space Search Engine for Web Services", in Proc. of the Third European Conference on Web Services (ECOWS' 05), Vaxjo, Sweden.

[15]. Porter, M. F. (1980) "An Algorithm for suffix Stripping", Program, vol. 14 (3), pp. 130-137.

[16]. Seretan V., Nerima L., Vehrli E. (2004) "Using the Web as a Corpus for the Syntactic-Based Collocation Identification", in Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 953-960, Sydney, Australia.

[17]. Trabalka M., Bieliková M. (2000) "Using XML and Regular Expressions in the Syntactic Analysis of Inflectional Language", In Proc. of Symposium on Advances in Databases and Information Systems (ADBIS-DASFAA'2000), Praha. pp. 185-194

[18]. Witten I. H., Frank E. (2005) "DataMining – Practical Machin Learning Tools and Techniques", 2nd ed., San Francisco: Morgan Kaufmann Publishers.

JBLIF, a Tool for Non-interference Analysis of Java and Java Bytecode Programs

Salvador V. Cavadini
Project EVEREST, Institut National de Recherche en Informatique et Automatique (INRIA)
Sophia-Antipolis, 06902, France
Salvador.Cavadini@sophia.inria.fr

Abstract

Protecting sensitive information has become an important facet of software development. One aspect of software security relies on information flow control (IFC), a technique for discovering information leaks in software. Despite the large body of work on language-based IFC, there are only few implementation of information flow analyzers for full-scale real programming languages. This lack signifies a gap between IFC theory and practice. This work introduces, a tool that helps to overpass this gap: JBLIF –acronym from Java Bytecode-Level Information Flow–, a tool capable of statically detect information leaks in systems coded in Java and/or Java bytecode.

Keywords: information flow control, non-interference, data security, software engineering.

1 Introduction

There is an urgent need for software applications with strong confidentiality guarantees. Protecting sensitive information –e.g. credit card data, personal medical information, military secrets– has become an important facet of software development. The problem is not new but it has acquired relevance due to ubiquity of computing systems.

One aspect of software security relies on *information flow control* (IFC), a technique for discovering information leaks in software. One of the two main tasks of IFC is guarantee that confidential data is not made public through public variables. Contemporary IFC use different kinds of program analysis to provide such a guarantee.

Language-based IFC uses the program code to discover security leaks. Most language-based IFC approaches uses non standard type systems where security levels are coded as special types for variables and the typing rules catch illegal flow of information [12].

Non-interference, a semantical condition on programs, ensures that high-security data will not be observable on low-security channels [3]. Despite the large body of work on language-based IFC, there are only few full-scale implementation of non-interference analyzers for real programming languages. This lack signifies a gap between theory and practice.

We have developed a tool that can help to overpass this gap: JBLIF –acronym from Java Bytecode-Level Information Flow–, is a tool capable of perform static non-interference analysis of software systems coded in Java and/or Java bytecode and supports security annotations at both levels: source and bytecode.

This paper is organized as follows. Next section explains how non-interference and program dependencies are related and why program slicing can be naturally used in non-interference analysis. Section 3 provides a description of JBLIF and section 4 shows two examples of non-interference analysis using JBLIF. Related work is discussed at section 5. Finally, the conclusions and future works.

2 Non-interference and Program Dependencies

Typically, a confidentiality policy labels certain variables as being secret to enforce the independence between the final value of non-secret –i.e. public– variables and the initial values of secret ones. This is semantically interpreted by *non-interference*: a program satisfy the confidentiality policy if every pair of computations, from a pair of initial states differing only in secret variables, leads to final states with identical public variables [3]. Non-interference generalizes to a security lattice with more than two elements but for the sake of the explanation two elements lattices are used.

The property of non-interference is naturally related with the dependencies between program statements, and expressions [1]. If statement y uses a variable defined at statement x, then y is *data dependent* on x. If the execution of statement y is controlled by the value of an expression x, then y is *control dependent* on x. The set of statements/expressions on whom y depends is called the *backward slice* of y [16]:

$$BS(y) = \{x | y \text{ depends on } x\}$$

From this definition, it is possible to conclude that if statement y directly or indirectly depends on statement x, then information could flow from x to y, noted $x \rightsquigarrow y$. If y do

not depends on x, then it is guaranteed that information can not flow from x to y, noted $x \not\rightsquigarrow y$ [14], symbolically:

$$x \notin BS(y) \implies x \not\rightsquigarrow y$$

2.1 Using Program Slicing in Non-interference Analysis

This last implication permits to develop a slicing-based non-interference analysis [5]. The idea is to mark certain selected statements as *providing* or *allowing* flows at certain security level. A *provided security level* specifies that a statement generates information at this security level. An *allowed security level* specifies that a statement accepts flows with a security level up to this security level.

Non-interference analysis consists in check that statements allowing security level l_1 do not depend on statements providing information at security level l_2 higher than l_1. More formally, program Prg is non-interferent iff

$$\forall a \in A : (\nexists p \in P \mid p \in BS(a) \text{ and } Allows(a) < Provides(p))$$

where A (P) is the set of statements in Prg with an allowed (provided) security level and $Allows(x)$ ($Provides(x)$) is the security level allowed (provided) by statement x.

The advantage of this approach is that non-interference analysis depends solely on the soundness property of correct slices: program slices are computed conservatively thus they may contain too many statements but never too few. Another advantage is that analysis precision depends on the slicing algorithm precision, permitting fine tuning of the computational resources to be used in non-interference analysis.

3 Non-interference Analysis at Java and Bytecode Levels with JBLIF

In a context where the interest in the enforcement of software security properties increases and the access to technologies allowing easy of information exchange –such as Internet and mobile phones– is generalized, tools able to check the security of mobile or embedded code-based platforms are of capital importance.

These kind of applications are mainly developed in Java language and usually available as bytecode. That is why it is important to have a security enforcement tool capable to work on Java and bytecode programs. JBLIF is such a tool. It can be used to enforce non-interference while system development and also to check that after compilation to bytecode the system still satisfies non-interference –thus untrusted Java compilers can be used–. Because JBLIF can work directly on bytecode, non-interference can be also checked when original Java sources are not available, a common case when the program is downloaded from Internet.

JBLIF is based on the ideas described in section 2, i.e. it performs non-interference analysis using the different dependencies between program sentences. More specifically, in JBLIF, non-interference analysis relies on program slices computed at bytecode level.

JBLIF is coded in Java and uses the slicing library of Indus [10], a framework for analysis of full Java programs.[1] Indus provides a set of libraries that works on Jimple [15]

[1] With the exception of dynamic class loading, reflection, and native methods.

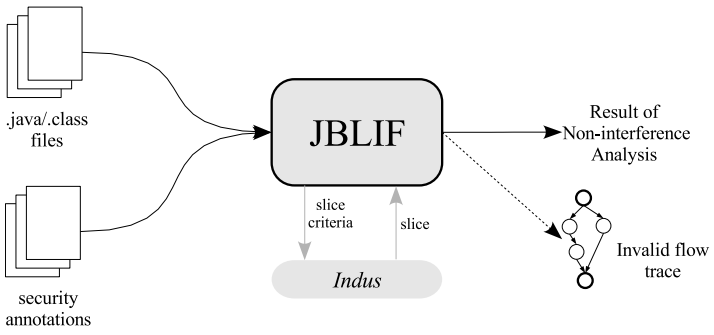


Figure 1: Highlevel architecture of JBLIF.

representation for Java bytecode. The slicing library gives access to functions computing highly customizable interprocedural context-sensitive slices in forward and backward direction.

In JBLIF, non-interference analysis is performed as follows –Figure 1–. First, the target system and the corresponding security annotations are loaded. Then, the tool computes the backward slice for each *allows-statement* and checks if some *provides-statement* in the slices has assigned a security level greater than the security level of the *allows-statement* being analyzed. If some invalid flow is detected, JBLIF informs it and generates a dependence graph that encodes the flows traces from the offending *provides-statement* to the *allows-statement*. The graph is the chop [7] between the two conflicting statements and it is generated in a format compatible with GraphViz [2].

3.1 Security Annotations in JBLIF

One characteristic that distinguish JBLIF from other non-interference analyzers is its ability to handle security annotations in both high level –Java– and low level –bytecode– programming languages. This way, users can add annotations to program components –variables, parameters, fields, statements, etc.– at the more convenient level.

Annotations are provided in separate files, one for each class in the system that the user needs to annotate and the same file can contain annotations at different levels. This scheme of annotations separated from the source code permits to check non-interference without modifying system files, thus preventing for unintentional program errors that could be introduced in the annotation process.

4 Examples

In this section, two small Java programs –listings 1 and 2– are used to illustrate how non-interference analysis is done with JBLIF.[2]

[2]Security annotations are given as program comments for the sake of clarity. As mentioned before, in JBLIF, security annotations are actually written in a separated text files.

Listing 1: A secure Java program.

```
1  public int secure (){
2
3      int secretInfo;
4      int publicInfo; // Provides: Public
5
6      secretInfo = System.in.read(); // Provides: Secret
7      if (secretInfo == 0){
8          publicInfo = 0
9      } else {
10         publicInfo = 1
11     };
12     publicInfo = 2;
13     System.out.print(publicInfo); // Allows: Public
14 }
```

The examples will also permit to emphasize that the dependence-based approach to non-interference analysis is more precise than the type system-based approach.

4.1 First Example: a simple Java program

This example exposes the main advantage of JBLIF non-interference analysis w.r.t. the type system based approach. Because type systems are usually flow insensitive they reject the program at Listing 1 considering that assignments at lines 8 and 10 are implicit flows from $secretInfo$ affecting the final value of $publicInfo$. By the contrary, JBLIF accepts the program because it can see that invalid implicit flows are killed by the assignment at line 12, thus the final value of $publicInfo$ is not related with $secretInfo$. More precisely, JBLIF computes the backward slice of sentence 13 –annotated as allowing up to *Public* level information flows–. This backward slice is the set of statements $\{4, 12, 13\}$ where only statement 4 is a *provides-statement* and generates a flow of *Public* information, thus the flow to statement 13 is valid.

If statement 12 is removed from the program, then the backward slice of 13 will result in the set $\{3, 6, 7, 8, 10\}$. Because statement 6 provides information at *Secret* level, JBLIF will be rejected the program as insecure.

4.2 Second Example: a more complex Java program

The program at Listing 2 –with a security lattice $High \rightarrow Low$– will permit to show how JBLIF deals with more complex features of Java language such as object sensitivity and dynamic dispatch. As explained before, JBLIF computes the backward slice for each one of the *allows-statements*, in this program, these statements are 18, 22, and 24. Then JBLIF checks each one of the slices looking for statements providing information at a security level higher than the allowed level:

Listing 2: A Java program with insecure flows (Taken from [4])

```
1  public class A {
2      int x;
3      void set(){x=0;}
4      void set(int i){x=i;}
5      int get() {return x;}
6  }
7  public class B extends A {
8      void set(){x=2;}
9  }
10 public class InfFlow {
11     public static void main(String[] a) {
12         int sec = 0; // Provides: High
13         int pub = 1; // Provides: Low
14         A o = new A();
15         o.set(sec);
16         o = new A();
17         o.set(pub);
18         outputInt(o.get()); // Allows: Low
19         if (sec==0 && a[0].equals("007"))
20             o = new B();
21         o.set();
22         System.out.println(o.get());// Allows: Low
23         o.set(42);
24         System.out.println(o instanceof B);// Allows: Low
25     }
26 }
```

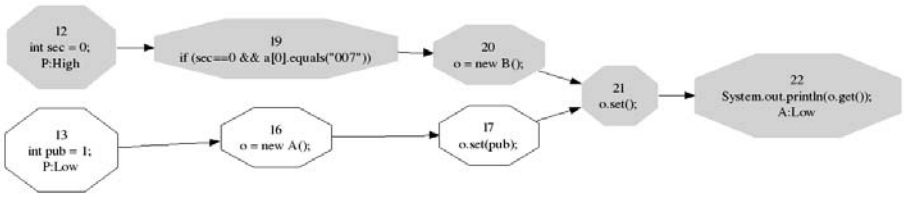


Figure 2: Partial backward slice, as is generated by JBLIF, from line 22 of Listing 2 highlighting –gray nodes– an invalid flow trace.

- The backward slice for statement 18 includes only one *provides-statement*: statement 13. Because 13 provides *Low* flows, the flow to 18 is safe. Notice that JBLIF is able to see that the object referenced at line 18, is in fact the object created at line 16 and set with a *Low* value –variable *pub*– and not the first created object –line 14– which was set with a $High$ value –variable *sec*–. This is possible because JBLIF has object sensitivity, something that is very hard to achieve with a type system.

- The backward slice for statement 22 includes statement 20, thus statements 19 and 12 are also included. Beacause the later provides a $High$ flow, JBLIF will indicate the existence of an invalid flow from 12 to 22, and the corresponding flow trace –Figure 2– is generated.

- The backward slice for statement 24, as the slice for 22 does, includes statement 12 because statements 19 and 20 are also included; JBLIF will highlight the invalid flow from 12 to 24 and generate the corresponding flow trace.

5 Related Work

Language-based information-flow security has a long, rich history with many –mostly theoretical– results [12]. Despite this large body of work, there are only few full language implementations of non-interference analyzers. In this section we provide a short overview of them and a comparison with JBLIF.

Jif is an information-flow typed extension of Java that builds upon the *decentralized label model* [9] and supports flexible and expressive information flow policies. Recently, Hicks et al. introduced FJifP, an extension of Jif that implements the *trusted declassification model* [6]. Flow Caml by Simonet et al. consists in an extension of the Objective Caml language with a type system tracing information flow [13].

The main advantage of JBLIF over these tools is that non-interference is enforced without rewriting the system in a new language. Moreover, the original source code of the system is not needed because JBLIF works at bytecode level, thus only `.class` files

are necessary. Other distinctive characteristic of JBLIF w.r.t. the above mentioned tools is related with the analysis approach: while JBLIF uses a dependence-based technique, Jif, FJifP, and Flow Caml use type systems; thus JBLIF analyses are more precise and, as mentioned previously, less false alarms are generated.

Recently, in [8], Li and Zdancewic presented an embedded security sublanguage of Haskell for enforcing information-flow policies in the standard Haskell programming language. Their approach has the advantage, over other systems like Jif, that the information-flow type system encoding is done using general features of Haskell without the need of a new language. Anyway, parts of the system where non-interference is to be enforced must be recoded, something that is not needed with JBLIF.

In [4], Hammer et al. present a dependence graphs-based system to check intransitive non-interference [11]. They inform that the system was implemented as an Eclipse plug-in that handles full Java. As far as we know, this tool and JBLIF are the sole tools implementing non-interference analysis for full Java language. The principal differences between JBLIF and this system are:

1. JBLIF works at both Java and bytecode level while Hammer's system works only at Java level,

2. JBLIF is not able to handle intentional information declassification as Hammer's system does,

3. In JBLIF, annotations are made in separated files and source code remains untouched. In Hammer's system, annotations are made in the Java source files,

4. JBLIF is available upon request to the authors.

6 Conclusions and Future Work

Despite the urgent need for strong confidentiality guarantees, the large body of literature, and considerable attention from the research community, information-flow based enforcement mechanisms are not widely used. One reason for this is the lack of a full-scale implementation of non-interference analysis for popular programming languages. JBLIF helps to overpass this gap between research and practice by making available non-interference analysis for full Java and also for Java bytecode. Moreover, JBLIF is the first tool to provide a flexible annotation mechanism allowing annotations at both Java, and bytecode level in separated files keeping source code untouched.

We have planned to extend JBLIF as a plug-in for the Eclipse IDE to facilitate the use of JBLIF in real cases of confidential Java programs development. Future work also includes adding the possibility of information declassification. Many realistic systems need to declassify some kind of confidential information as part of their normal behavior. The actual challenge is to differentiate between proper and improper declassification of confidential information. This problem will be the object of our coming research efforts.

JBLIF is available upon request to the authors.

References

[1] Martín Abadi, Anindya Banerjee, Nevin Heintze, and Jon G. Riecke. A core calculus of dependency. In *POPL '99. Proceedings of the 26th ACM SIGPLAN-SIGACT on Principles of programming languages, January 20–22, 1999, San Antonio, TX*, pages 147–160, New York, NY, USA, 1999. ACM Press.

[2] Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software — Practice and Experience*, 30(11):1203–1233, 2000.

[3] Joseph A. Goguen and José Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, pages 11–20, 1982.

[4] Christian Hammer, Jens Krinke, and Frank Nodes. Intransitive noninterference in dependence graphs. In *Second International Symposium on Leveraging Application of Formal Methods, Verification and Validation (ISoLA 2006)*, pages 136–145, 2006.

[5] Christian Hammer, Jens Krinke, and Gregor Snelting. Information flow control for java based on path conditions in dependence graphs. In *IEEE International Symposium on Secure Software Engineering*, 2006.

[6] Boniface Hicks, Dave King, Patrick McDaniel, and Michael Hicks. Trusted declassification:: high-level policy for a security-typed language. In *PLAS '06: Proceedings of the 2006 workshop on Programming languages and analysis for security*, pages 65–74, New York, NY, USA, 2006. ACM Press.

[7] Daniel Jackson and Eugene J. Rollins. A new model of program dependences for reverse engineering. In *SIGSOFT '94: Proceedings of the 2nd ACM SIGSOFT symposium on Foundations of software engineering*, pages 2–10, New York, NY, USA, 1994. ACM Press.

[8] Peng Li and Steve Zdancewic. Encoding information flow in haskell. In *CSFW '06: Proceedings of the 19th IEEE Workshop on Computer Security Foundations*, page 16, Washington, DC, USA, 2006. IEEE Computer Society.

[9] Myers and Liskov. Complete, safe information flow with decentralized labels. In *RSP: 19th IEEE Computer Society Symposium on Research in Security and Privacy*, 1998.

[10] V. P. Ranganath and J. Hatcliff. An overview of the indus framework for analysis and slicing of concurrent java software (keynote talk - extended abstract). pages 3–7, 2006.

[11] A. W. Roscoe and Michael Goldsmith. What is intransitive noninterference? In *PCSFW: Proceedings of The 12th Computer Security Foundations Workshop*. IEEE Computer Society Press, 1999.

[12] A. Sabelfeld and A. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications, 21(1)*, 2003.

[13] V. Simonet. Flow caml in a nutshell. In *Proceedings of the first APPSEM-II workshop*, pages 152–165, 2003.

[14] Gregor Snelting, Torsten Robschink, and Jens Krinke. Efficient path conditions in dependence graphs for software safety analysis. 15(4):410–457, October 2006.

[15] Raja Vallee-Rai and Laurie J. Hendren. Jimple: Simplifying java bytecode for analyses and transformations, 1998.

[16] Mark Weiser. Program slicing. In *ICSE '81: Proceedings of the 5th international conference on Software engineering*, pages 439–449, Piscataway, NJ, USA, 1981. IEEE Press.

Listas de Clusters usando Centros Espacialmente Dispersos para Búsquedas por Similaridad en espacios Métricos.

Roberto Uribe-Paredes[1,2], Roberto Solar[1], and Claudio Márquez[1]

[1] Depto. de Ingeniería en Computación*,
Universidad de Magallanes, Chile
[2] Grupo de Bases de Datos - UART,
Universidad Nacional de la Patagonia Austral, Río Turbio, Argentina
E-mail: ruribe@ona.fi.umag.cl

Abstract. *Sparse Spatial Selection* is a new pivot-based structure for similarity search in metric spaces. It is a array-type structure which has shown a good search performance when compared with other selection methods.
This work describes the building of a new metric structure. It is a tree-type structure and born as a recursive application from *Cluster List* and using *SSS* as a general method for center selection. Preliminaries experimental results show that it presents a better performance, in terms of the number of distance evaluations than original *Cluster List* and other well-known structures.
Keywords: databases, data estructures, algorithms, metric spaces, similarity queries.

Resumen El *Sparse Spatial Selection* es una nueva estructura basada en pivotes para búsqueda por similaridad en espacios métricos. Esta estructura es del tipo arreglo y ha demostrado buen rendimiento durante la búsqueda comparado con otros métodos de selección.
El presente trabajo describe la construcción de una nueva estructura métrica. Ésta es una estructura de tipo árbol y nace de la aplicación recursiva de *Listas de Clusters* usando *SSS* como un método general para la selección de centros. Resultados experimentales preliminares demuestran que tiene mejor desempeño, en términos de evaluaciones de distancia, que la estrutura *Lista de Clusters* original y otras estructuras conocidas.
Palabras claves: bases de datos, estructuras de datos, algoritmos, espacios métricos, consultas por similaridad.

1. Introducción

1.1. Antecedentes

Uno de los problemas de gran interés en ciencias de la computación es el de "búsqueda por similaridad", es decir, encontrar los elementos de un conjunto más similares a una muestra. Esta búsqueda es necesaria en múltiples aplicaciones, como ser en reconocimiento de voz e imagen, compresión de video, genética, minería de datos, recuperación de información, etc. En casi todas las aplicaciones la evaluación de la similaridad entre dos elementos es cara, por lo que usualmente se trata como medida del costo de la búsqueda la cantidad de similaridades que se evalúan.

Interesa el caso donde la similaridad describe un espacio métrico, es decir, está modelada por una función de distancia que respeta la desigualdad triangular. En este caso, el problema más común y difícil es en aquellos espacios de "alta dimensión" donde el histograma de distancias es concentrado, es decir, todos los objetos están más o menos a la misma distancia unos de otros.

El aumento de tamaño de las bases de datos y la aparición de nuevos tipos de datos sobre los cuales no interesa realizar búsquedas exactas, crean la necesidad de plantear nuevas estructuras para búsqueda por similaridad o búsqueda aproximada, esto en busca de superar los problemas de las

* Parcialmente financiado por los proyectos Fondecyt 1060776, Conicyt; programa de investigación PR-F1-002IC-06, Universidad de Magallanes, Chile.

estructuras existentes hasta ahora. Asimismo, se necesita que dichas estructuras sean dinámicas, es decir, que permitan agregar o eliminar elementos sin necesidad de crearlas nuevamente. Así también, las aplicaciones reales requieren que dichas estructuras permitan ser almacenadas en memoria secundaria eficientemente, como también que posean métodos optimizados para reducir los costos de accesos a disco. Adicionalmente, el método elegido en las estructuras para la elección de centros y pivotes resulta relevante al momento de descartar elementos durante la búsqueda.

1.2. Marco teórico

La similaridad se modeliza en muchos casos interesantes a través de un espacio métrico, y la búsqueda de objetos más similares a través de una búsqueda por rango o de vecinos más cercanos.

Definición 1 (*Espacios Métricos*): Un espacio métrico es un conjunto X con una función de distancia $d : X^2 \rightarrow R$, tal que $\forall x, y, z \in X$,

1. $d(x, y) \geq 0$ *and* $d(x, y) = 0$ *ssi* $x = y$. *(positividad)*
2. $d(x, y) = d(y, x)$. *(Simetría)*
3. $d(x, y) + d(y, z) \geq (d(x, z)$. *(Desigualdad Triangular)*

Definición 2 (*Consulta por Rango*): Sea un espacio métrico *(X,d)*, un conjunto de datos finito $Y \subseteq X$, una consulta $x \in X$, y un rango $r \in R$. La consulta de rango alrededor de x con rango r es el conjunto de puntos $y \in Y$, tal que $d(x, y) \leq r$.

Definición 3 (*Los k Vecinos más Cercanos*): Sea un espacio métrico *(X,d)*, un conjunto de datos finito $Y \subseteq X$, una consulta $x \in X$ y un entero k. Los k vecinos más cercanos a x son un subconjunto A de objetos de Y, donde la $|A| = k$ y no existe un objeto $y \in A$ tal que *d(y,x)* sea menor a la distancia de algún objeto de A a x.

El objetivo de los algoritmos de búsqueda es minimizar la cantidad de evaluaciones de distancia realizadas para resolver la consulta. Los métodos para buscar en espacios métricos se basan principalmente en dividir el espacio empleando la distancia a uno o más objetos seleccionados. El no trabajar con las características particulares de cada aplicación tiene la ventaja de ser más general, pues los algoritmos funcionan con cualquier tipo de objeto [9].

Existen distintas estructuras para buscar en espacios métricos, las cuales pueden ocupar funciones discretas o continuas de distancia. Algunos son BKTree [4], MetricTree [15], GNAT [2], VpTree [17], FQTree [1], MTree [10], SAT [12], Slim-Tree [14], EGNAT [16].

Algunas de las estructuras anteriores basan la búsqueda en pivotes y otras en clustering. En el primer caso se seleccionan pivotes del conjunto de datos y se precalculan las distancias entre los elementos y los pivotes. Cuando se realiza una consulta, se calcula la distancia de la consulta a los pivotes y se usa la desigualdad triangular para descartar candidatos.

Los algoritmos basados en clustering dividen el espacio en áreas, donde cada área tiene un *centro*. Se almacena alguna información sobre el área que permita descartar toda el área mediante sólo comparar la consulta con su centro. Los algoritmos de clustering son los mejores para espacios de alta dimensión, que es el problema más difícil en la práctica.

Existen dos criterios para delimitar las áreas en las estructuras basadas en clustering, *hiperplanos* y *radio cobertor* (*covering radius*). El primero divide el espacio en particiones de *Voronoi* y determina el hiperplano al cual pertenece la consulta según a qué centro corresponde. El criterio de radio cobertor divide el espacio en esferas que pueden intersectarse y una consulta puede pertenecer a más de una esfera.

Definición 4 (*Diagrama de Voronoi*): Considérese un conjunto de puntos $\{c_1, c_2, \ldots, c_n\}$(centros). Se define el diagrama de Voronoi como la subdivisión del plano en n áreas, una por cada c_i, tal que $q \in$ al área c_i sí y sólo sí la distancia euclidiana $d(q, c_i) < d(q, c_j)$ para cada c_j, con $j \neq i$.

Uno de los problemas que provoca que muchas veces buenas estructuras arrojen malos resultados es la elección poco afortunada de centros o pivotes. En este sentido, este trabajo propone el uso de un nuevo método denominado *Sparse Spatial Selection* (*SSS*) [3,13], como un método general para selección de centros o pivotes. Para la aplicación de este método, se eligió inicialmente la estructura denominada *Lista de Clusters* [8], que es de las estructuras basadas en clustering, del tipo arreglo y utiliza el radio cobertor para descartar centros durante la búsqueda.

Para este artículo se seleccionó, para la realización de las pruebas, un espacio métrico consistente en un diccionario de palabras en castellano de 86.061 objetos, donde la distancia utilizada es la *distancia de edición*, la cual entrega como resultado el número mínimo de inserciones, eliminaciones o reemplazos de caracteres, necesarios, para que una palabra sea igual a otra. El segundo es un espacio de vectores de coordenadas reales de dimensión 10 generados con distribución de *Gauss* con media 1 y varianza 0.1 cuya cantidad de objetos es de 100,000, para este espacio se utilizó la *distancia Euclidiana*. Para la búsqueda se creó la estructura con el 90 % de los datos y se reservó el 10 % como consultas.

2. Selección de Pivotes y Centros

En particular, la elección de pivotes o centros según sea la estructura resulta relevante para obtener un mayor rendimiento durante la búsqueda, lo que queda demostrado empíricamente en [5].

Distintas estrategias han sido propuestas como adecuadas para la elección de pivotes. En [11] se propone seleccionar como pivotes aquellos objetos que maximicen la suma de las distancias a los pivotes ya seleccionados. En [2,17] se siguen heurísticas que tratan de seleccionar pivotes que estén lejanos entre sí. En [5] se presenta un criterio de comparación de la eficiencia entre dos conjuntos de pivotes, así también se presentan varias estrategias de selección de conjuntos donde se usa el criterio de eficiencia anterior.

El método denominado *Sparse Spatial Selection* (*SSS*), es una nueva técnica propuesta para la selección de pivotes, ésta fue originalmente implementado sobre una estructura del tipo arreglo y usa la desigualdad triangular para discriminar objetos durante la búsqueda. Dicho método se comportó igual o mejor en términos de eficiencia que los propuestos en [5], con la ventaja adicional que no se requiere determinar a priori el número de pivotes necesarios.

2.1. *Sparse Spatial Selection (SSS)*

SSS es un método de selección de pivotes en el que escoge un conjunto dinámico de pivotes bien distribuidos en el espacio, lo que permite descartar más objetos al momento de realizar una búsqueda.

Sea $(\mathbb{X}, d)$ un espacio métrico, $\mathbb{U} \subset \mathbb{X}$ una colección de objetos y M la distancia entre los dos objetos más alejados. En un principio el conjunto de pivotes está formado por el primer objeto de la colección. Luego, para cada elemento de la colección se verifica si está a una distancia mayor o igual a $M * \alpha$ de los pivotes seleccionados, si es así, se agrega al conjunto de pivotes, siendo α una constante cuyos valores están cercanos a 0,4[13]. La figura 1 muestra la obtención de los pivotes en un espacio cualquiera.

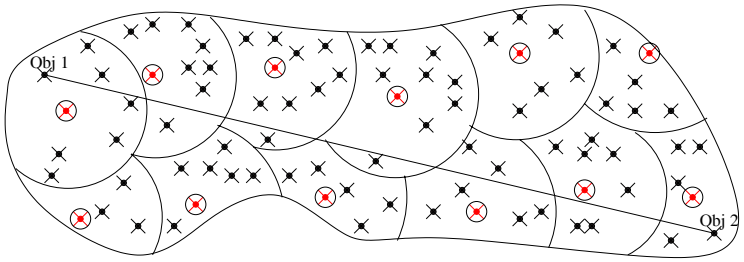


Figura 1. Los objetos encerrados en círculos representan los centros seleccionados como pivotes y M se define como la distancia entre Obj 1 y Obj 2, los más alejados en el espacio.

En [13] la estructura original es basada en pivotes y del tipo arreglo. Su construcción es similar a las estructuras *FQA*[7] y *Spaghettis*[6], pero se diferencia en la forma de elegir los pivotes y en la manera de buscar. Básicamente se tiene un arreglo donde la cantidad de filas es la cantidad total de objetos en la Base de Datos y la cantidad de columnas es el número de pivotes.

En el presente trabajo se considera que el *SSS* es básicamente un método de selección de pivotes, por lo que puede ser aplicado a cualquier estructura, independiente del tipo y de los criterios para delimitar áreas. Se considera también, que es posible construir una estructura plenamente basada en el *SSS*, es decir, una estructura que puede ajustarse completamente al espacio métrico sobre el cual es implementada.

3. *Lista de Clusters*

La ***Lista de Clusters*** [8]es una estructura basada en Clustering o particiones compactas, la cual es muy similar a una Lista Enlazada. Diseñada para tener un buen desempeño en espacios de altas dimensiones.

En la *Lista de Clusters* se selecciona un centro c perteneciente a la base de datos X y un radio r el cual determina la fracción del espacio que abarca la esfera (c, r) definida como el subconjunto de elementos de X los cuales están a una distancia no mayor a r del centro c. Luego se define como I a los elementos que están dentro de la esfera de centro c también llamado *Bucket*, y E definido como el resto de los elementos externos a la esfera de centro c. Este proceso se repite recursivamente. En consecuencia se obtiene una lista compuesta por un centro, un radio y un Bucket (c,r,I) denominado Cluster.

Comparado con otros algoritmos de Clustering, la *Lista de Clusters* solamente usa el criterio de radio cobertor y no áreas como en el Voronoi-Tree. También es posible ver la *Lista de Clusters* como un caso particular de Voronio-tree o un M-tree, considerando I y E como los sub-árboles izquierda y derecha de raíz c, con las diferencias de que las estructuras recién mencionadas tratan de construir un árbol balanceado y que además poseen estructuras internas, en cambio la *Lista de Clusters* es extremadamente desbalanceada y no poseen estructura interna alguna.

3.1. Construcción

La estructura *Lista de Clusters* se construye según el algoritmo 1, donde U que corresponde a los datos no insertados en la lista:

Algoritmo 1 $ContruirLC(U)$

if $U = \phi$ **then**
  retorna una lista vacia
end if
Se Selecciona $c \in U$
Se Selecciona un radio r
$I \longleftarrow \{\ u \in U - c\ ,\ d(c,u) \leq r\ \}$
$E \longleftarrow U - I$
retorna (c, r, I)

La estructura de datos construida debería ser simétrica, pero no lo es. El primer centro escogido tiene preferencia sobre los centros subsecuentes por lo que se provoca solapamiento entre clusters. La figura 2 lo ilustra. Todos los elementos que están dentro del cluster del primer centro (c1 en la figura) se guardan en su Bucket I, a pesar de eso ellos pueden quedar también dentro de los Buckets I de centros subsecuentes (c2, c3, etc. figura 2).

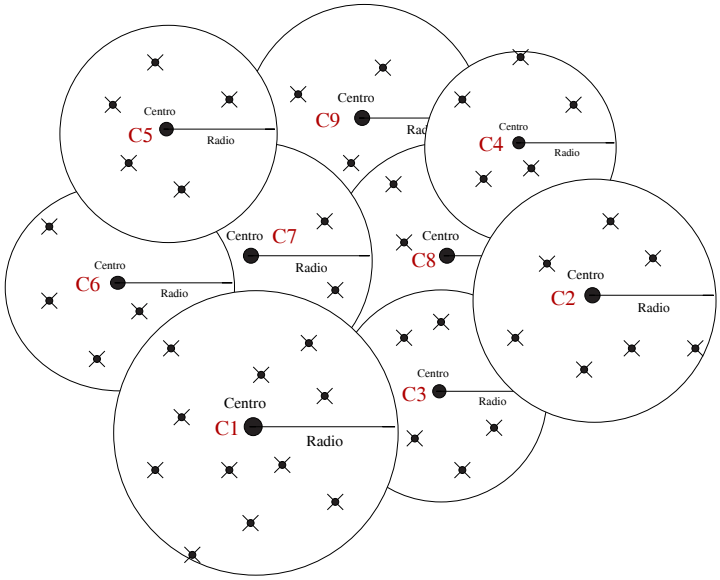


Figura 2. Zonas de influencia de 9 Clusters construidos según el orden : C1, C2, C3, C4, C5, C6, C7, C8 y C9.

3.2. Búsqueda

Dado la asimetría de esta estructura, también la búsqueda se puede reducir si la consulta esta totalmente contenida en el Cluster, por lo cual no se necesita considerar E porque por construcción todos los elementos que están dentro de la esfera de consulta han sido insertados en I. La búsqueda se muestra en el algoritmo 2, en el cual se aplica una consulta q y un radio de búsqueda r sobre la Lista de Cluster L :

Algoritmo 2 *BusquedaLC*(L,q,r)

if L *es vacia* **then**
  Se retorna
end if
$L = (c, r_c, I)$
Se Calcula $d(c, q)$
if $d(c, q) \leq r$ **then**
  c *es un resultado*
end if
if $d(c, q) \leq r_c + r$ **then**
  Se debe buscar exhaustivamente en I
end if
if $d(c, q) > r_c - r$ **then**
  Se debe seguir buscando en (E, q, r)
end if

Esta es una característica esencial ausente en otros algoritmos de Clustering, donde la búsqueda necesita entrar en todos los Clusters que son intersectados por la esfera de consulta. En esta estructura la búsqueda sobre los Cluster restante puede ser cancelada en cuanto la esfera de consulta este totalmente contenida en un Cluster. En la figura 3 se puede apreciar tres casos de consultas sobre un Cluster, en el caso de $q1$ se debe considerar el Bucket actual y el resto de los Clusters, para el caso de $q2$ se debe hacer la búsqueda sólo en este Bucket y para $q3$ evitamos la búsqueda en el Bucket actual.

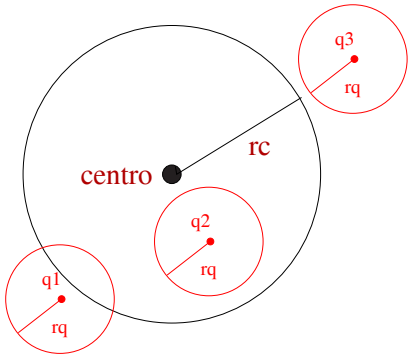


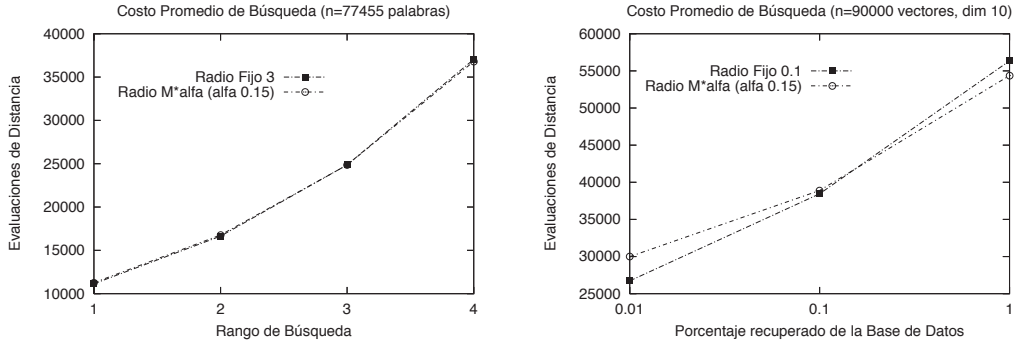
Figura 3. Tres casos de consultas sobre un centro de Cluster.

En las características generales de esta estructura no se especifica cómo se seleccionan los centros y radios en cada punto del algoritmo de construcción, ya que esto se relaciona con la eficacia y no a la exactitud de la estructura de datos. Una buena selección de centros podría ser más bien costosa, además debemos tomar en cuenta que se debe seleccionar un buen radio para cada centro de cluster en cada punto de la construcción debido a que el espacio va cambiando en la medida que los objetos son insertando en la Lista. En este sentido resulta interesante aplicar el método SSS para la selección de centros versus la opción de selección aleatoria. La estructura posee dos alternativas de construcción, la primera es seleccionar un radio fijo para todos los Clusters de la lista y la segunda es seleccionar un tamaño fijo para todos los Clusters de la lista, *Cluster de Radio Fijo* y *Cluster de Tamaño Fijo* respectivamente.

4. *Lista de Clusters* y *Sparse Spatial Selection*

Durante la construcción, la *Lista de Cluster* selecciona inicialmente un centro y un radio, por lo que resulta natural elegir como radio $M * \alpha$ y cada centro usando el algoritmo SSS, es decir, los centros son elegidos si están ubicados a una distancia $M * \alpha$ de todos los centros anteriores. SSS puede ser utilizado para radio fijo, como para tamaño fijo, sin embargo, pruebas preliminares determinaron un comportamiento superior en listas de cluster con radio fijo. El cálculo del valor de M se realiza sobre todos los objetos de la base de datos, sin embargo, esto es un proceso off-line y en este trabajo no se lo consideró como costo de construcción.

La figura 4 muestra el comportamiento de la estructura durante la búsqueda. Los gráficos son para la estructura Lista de Clusters en su versión de radio fijo versus la alternativa de selección de centros usando SSS y radio $M * \alpha$. En ambos casos, los gráficos corresponden a los mejores valores encontrados para radio fijo y a los mejores valores para α. Se puede observar que la diferencia es ínfima, sobre todo en el caso del espacio de palabras, ello es debido a que la función de distancia es discreta, por lo que el mejor radio, es muy parecido al mejor α. En el caso del espacio de vectores, existe una pequeña mejora al aumentar los rangos de búsqueda, en este experimento, el mejor radio es aquel que recupera el 0,1 % de los datos.



(a) Diccionario Español (b) Gauss Dimensión 10

Figura 4. Costos promedios de Búsqueda Lista de Cluster de Radio Fijo vs Lista de Cluster con SSS

4.1. *Lista de Cluster* Recursiva

En los gráficos mostrados anteriormente no se logra ver una mejora importante en el desempeño de la estructura. Esto resulta así, dado que se seleccionaron los mejores métodos para ambos experimentos. El mejor radio fijo y α fueron obtenidos experimentalmente.

Considerando que el espacio es dividido una sola vez en N partes, es posible aplicar esa división en los subespacios generados, es decir, cada cluster de la estructura puede ser a su vez una *Lista de Clusters*. Entonces, una segunda alternativa de construcción es aplicar recursivamente el proceso de construcción sobre cada cluster formado en la estructura original. Para la construcción de esta estructura se utiliza el método SSS.

Finalmente, lo que se obtiene es una estructura de tipo árbol donde se seleccionan los centros espacialmente dispersos, usando un radio de $M * \alpha$.

Cada cluster de la estructura original representa un subespacio con características distintas al original, de hecho los rangos de dicho espacio son mucho menores. Esto puede ser considerado una ventaja, ya que al aplicar recursivamente el método SSS usando el M original, el espacio quedaría sobredimensionado, provocando baja en la eficiencia al interior de la estructura. Ahora, es posible calcular nuevamente el M para el nuevo subespacio, pero implicaría un costo demasiado elevado durante la construcción. Sin embargo, es posible utilizar el mismo radio cobertor del subespacio para calcular un M aproximado, sin pagar costos adicionales.

El radio cobertor es la distancia desde el centro a su elemento más alejado, por lo que se garantiza que M siempre será menor o igual a $2 * rc$ (dos veces el radio cobertor). Esto puede ser utilizado cada vez que se realiza el proceso de construcción recursivamente.

La utilización recursiva del método SSS sobre cada cluster, modificando el valor de M, provoca que la estructura se vaya adaptando siempre a la nueva forma del espacio. Lo anterior implica que la cantidad de nodos en el árbol será dinámica, es decir, los nodos usualmente no tendrán la misma cantidad de objetos.

Finalmente, el proceso termina cuando el cluster tiene una cantidad de datos inferior a una cota determinada, por ejemplo a una página de disco.

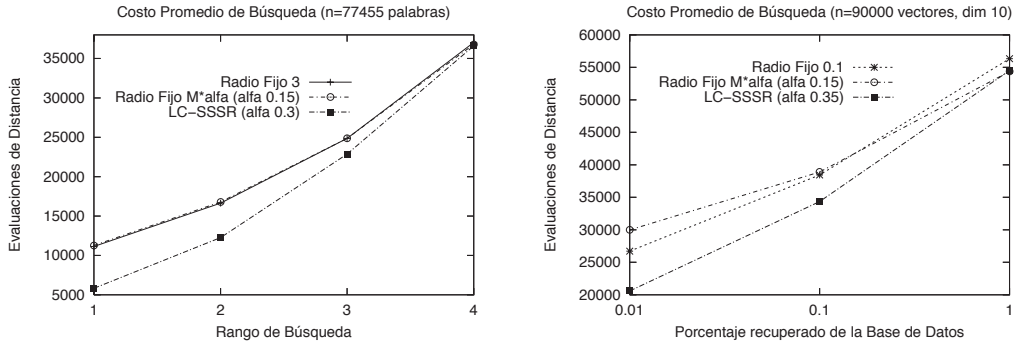
4.2. Resultados Experimentales

En las figuras 5 y 6 se pueden observar resultados interesantes sobre el comportamiento de la nueva estructura (identificada como *LC-SSSR)*. En el espacio de Gauss, los valores graficados corresponden a los rangos que permiten recuperar el 0,01, el 0,1 y el 1 %. La figura 5 corresponde a mejores versiones de Lista de Cluster de radio fijo y Lista de Cluster con SSS versus Lista de Cluster con SSS y Recursiva

LC-SSSR). En este experimento se puede ver claramente que la nueva estructura tiene una notable ventaja sobre las dos versiones anteriores, siendo dicha ventaja muy superior en los casos de bajos rangos de búsqueda.

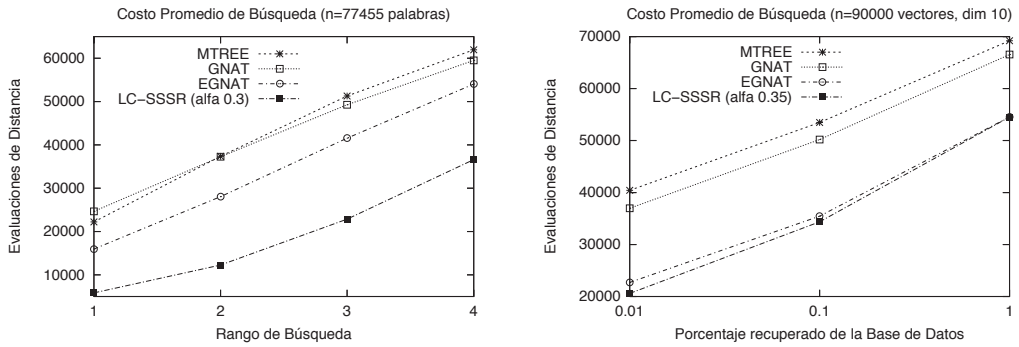
La figura 6 muestra los resultados para los mismos dos espacios de la nueva estructura versus tres estructuras conocidas de buen desempeño en espacios de alta dimensión. De los gráficos de búsqueda se puede ver claramente que el nuevo método aventaja a las estructuras *MTree*, *GNAT* y *EGNAT* muy notoriamente en el espacio de palabras. En el espacio de Gauss el *LC-SSSR* tiene un desempeño levemente mejor al *EGNAT*.

La figura 6 corresponde a un gráfico comparativo entre LC-SSSR y el EGNAT, la mejor de las estructuras comparadas. El experimento fue realizado sobre una nueva colección de datos, la que corresponde a un conjunto de 47.000 imágenes extraídas de archivos de la NASA y convertidas a vectores de 20 componentes. En esta colección, se puede notar que la estructura LC-SSSR aún mantiene una ventaja importante sobre la estructura EGNAT.



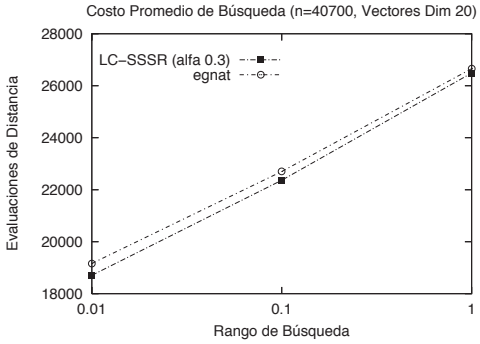
(a) Diccionario Español (b) Gauss Dimensión 10

Figura 5. Costos promedios de Búsqueda. Mejores versiones de Lista de Cluster versus la nueva estructura.



(a) Diccionario Español (b) Gauss Dimensión 10

Figura 6. Gráficos comparativos para la búsqueda (MTree v/s GNAT v/s EGNAT v/s LC-SSSR).



(a) Espacio de vectores de dimensión 20

Figura 7. Gráficos comparativos para la búsqueda (EGNAT v/s LC-SSSR).

5. Conclusiones

5.1. Aspectos Relevantes y Aportes

Una buena elección de pivotes y centros durante la construcción de estructuras métricas siempre será relevante para los procesos de búsqueda. Considerando que los mejores centros serán dependientes del espacio, es ideal contar con mecanismos que permitan recolectar, independiente de la forma del espacio, la mejores alternativas de centros.

En este sentido, los autores consideran que el método *SSS* permite, efectivamente, obtener un conjunto adecuado de centros, lo que queda demostrado claramente en el presente artículo.

Se considera que el principal aporte del presente trabajo es desarrollar la propuesta de una nueva estructura inicialmente denominada LC-SSSR, basada en la estructura *Lista de Clusters* con aplicación resursiva de la selección de centros espacialmente esparcidos (*SSS*). Esta estructura utiliza en forma natural el método *SSS* durante la construcción. Es basada en clustering y del tipo árbol y es competitiva en espacios de alta dimensión.

Los resultados experimentales demuestran lo anterior y proporcionan una visión de las enormes ventajas frente a otras estructuras que son prometedoras.

Es importante mencionar que la nueva estructura se va adaptando a la forma del espacio, dado el uso de SSS. Esto también es posible debido al recálculo del valor de M durante la construcción, lo que no tiene costos adicionales.

5.2. Trabajos Futuros

Los autores consideran que el desarrollo de esta estructura está aún en proceso de investigación y experimentación, por lo que se considera entre los trabajos futuros, la comparación con otras estructuras, la utilización de nuevas colecciones de objetos, la búsqueda de los mejores α's según cada subespacio o nivel interno de la estructura. Según los autores, una de las tareas pendientes es disminuir aún más los cálculos de distancia utilizando técnicas de descarte distintas al radio cobertor. Finalmente, determinar las capacidades dinámicas de la estructura y su comportamiento en memoria secundaria.

Referencias

1. R. Baeza-Yates, W. Cunto, U. Manber, and S. Wu. Proximity matching using fixedqueries trees. In *5th Combinatorial Pattern Matching (CPM'94)*, LNCS 807, pages 198–212, 1994.

2. Sergei Brin. Near neighbor search in large metric spaces. In *the 21st VLDB Conference*, pages 574–584. Morgan Kaufmann Publishers, 1995.
3. N. Brisaboa, A. Fariña, O. Pedreira, and N. Reyes. Selección espacial de pivotes dispersos para la búsqueda por similitud en espacios métricos. In *XII Congreso Argentino de Ciencias de la Computación*, Oct. 2006. San Luis, Argentina.
4. W. Burkhard and R. Keller. Some approaches to best-match file searching. *Communication of ACM*, 16(4):230–236, 1973.
5. B. Bustos, G. Navarro, and E. Chávez. Pivot selection techniques for proximity search in metric spaces. In *XXI Conference of the Chilean Computer Science Society*, pages 33–40. SCCC, IEEE Computer Science Press, 2001.
6. E. Chavéz, J. Marroquín, and R. Baeza-Yates. Spaghettis: An array based algorithm for similarity queries in metric spaces. In *6th International Symposium on String Processing and Information Retrieval (SPIRE'99)*, pages 38–46. IEEE CS Press, 1999.
7. E. Chavéz, J. Marroquín, and G. Navarro. Fixed queries array: A fast and economical data structure for proximity searching. *Multimedia Tools and Applications*, 14(2):113–135, 2001.
8. E. Chavéz and G. Navarro. An effective clustering algorithm to index high dimensional metric spaces. In *The 7th International Symposium on String Processing and Information Retrieval (SPIRE'2000)*, pages 75–86. IEEE CS Press, 2000.
9. Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José L. Marroquín. Searching in metric spaces. In *ACM Computing Surveys*, pages 33(3):273–321, September 2001.
10. P. Ciaccia, M. Patella, and P. Zezula. M-tree : An efficient access method for similarity search in metric spaces. In *the 23st International Conference on VLDB*, pages 426–435, 1997.
11. L. Micó, J. Oncina, and E. Vidal. A new version of the nearest-neighbor approximating and eliminating search (aesa) with linear preprocessing-time and memory requirements. *Pattern Recognition Letters*, 15:9–17, 1994.
12. Gonzalo Navarro. Searching in metric spaces by spatial approximation. *The Very Large Databases Journal (VLDBJ)*, 11(1):28–46, 2002.
13. Oscar Pedreira and Nieves R. Brisaboa. Spatial selection of sparse pivots for similarity search in metric spaces. In *SOFSEM 2007: 33rd Conference on Current Trends in Theory and Practice of Computer Science*, volume 4362 of *Lecture Notes in Computer Science*, pages 434–445, Harrachov, Czech Republic, January, 20-26 2007. Springer.
14. Caetano Traina, Agma Traina, Bernhard Seeger, and Christos Faloutsos. Slim-trees: High performance metric trees minimizing overlap between nodes. In *VII International Conference on Extending Database Technology*, pages 51–61, 2000.
15. J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. In *Information Processing Letters*, pages 40:175–179, 1991.
16. Roberto Uribe-Paredes. Manipulación de estructuras métricas en memoria secundaria. Master's thesis, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Santiago, Chile, Abril 2005.
17. P. Yianilos. Data structures and algoritms for nearest neighbor search in general metric spaces. In *4th ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, pages 311–321, 1993.

Modelado de Sistema Experto para Triaje en Servicios de Urgencias Médicas

María M. Abad-Grau
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada
E-mail: mabad@ugr.es
Tel: +34 958240634
Fax: +34 958243179

Jorge S. Ierache
ISIER - Facultad de Informática Ciencias de la Comunicación y Técnicas Especiales,
Universidad de Morón
jierache@unimoron.edu.ar
Tel: +5411 56272000 (189/746)

Claudio Cervino
Facultad de Medicina
Universidad de Morón
ccervino@unimoron.edu.ar
Tel: +5411 56272000 (742**)**

Abstract

The Probabilistic graphical models, such as the bayesian networks and the diagrams of influences allow to represent of coherent form the knowledge of a dominion under conditions of uncertainty. They are based on the foundations of the theory of the probability and allow to combine the judgment of the expert with the sources of data available. This article describes the present work that we are making for the application of bayesian networks in the modeled one of expert systems of triage (classification) in the services of medical urgencies. The networks are constructed considering as much the originating data of experiences of triage like the opinion of expert doctors in urgencies. The system will be used with one double purpose: at theoretical level to understand how the information required in the triage can be modeled by means of Bayesian networks and at practical level for training and use by the triage personnel.

Key words: Bioinformatics, Triage, Expert systems, Bayesian Networks

Resumen

Los modelos gráficos probabilísticos, tales como las redes bayesianas y los diagramas de influencias permiten representar de forma coherente el conocimiento de un dominio bajo condiciones de incertidumbre. Están basados en los fundamentos de la teoría de la probabilidad y permiten combinar el juicio del experto con las fuentes de datos disponibles. Este articulo describe el trabajo actual que estamos realizando para la aplicación de redes bayesianas en el modelado de sistemas expertos de triaje (clasificación) en los servicios de urgencias médicas. Las redes son construidas teniendo en cuenta tanto los datos provenientes de experiencias de triaje como la opinión de médicos expertos en urgencias. El sistema será utilizado con una doble finalidad: a nivel teórico para entender cómo la información requerida en el triaje puede ser modelada mediante redes bayesianas y a nivel práctico para entrenamiento y uso por el personal de triaje.

Palabras claves: Bioinformática, Triaje, Sistemas Expertos, Redes Bayesianas

1. INTRODUCCIÓN

Los servicios médicos de urgencias en la mayoría de los hospitales están sufriendo un fuerte incremento en el número de casos que se presentan. Existen muchos factores, tales como la frecuente imposibilidad de ser atendidos por familiares en el hogar que aumentan el número de pacientes con ingreso hospitalario. Con el fin de usar los recursos de forma más eficiente, cada vez en más hospitales se hace imprescindible la implantación de un sistema de triaje, o sistema para la catalogación de la urgencia con la que un paciente en la sala de emergencias debe ser atendido.

Algunos ejemplos de sistemas de implantación a gran escala [10], son el sistema canadiense CTAS (The Canadian Emergency Department Triage & Acuity Scale), el ESI (Emergency Severity Index), el MTS (Manchester Triage System) o el MAT (Modelo Andorrano de Triaje). Todos ellos usan una escala de categorías de 5 niveles (ver Tabla 1). Escalas de menos valores no han conseguido los mismos niveles de reproducibilidad, utilidad y validez suficientes para se considerados estándares [7]. De todos ellos, el CTAS, de amplia difusión en el continente americano y el MAT, implantado sobre todo en España, disponen de análisis de concordancia y estudios de validez y utilidad. MAT además se basa en categorías sintomáticas y algoritmos clínicos y se encuentra en versión electrónica.

En la sección 1 hacemos una revisión del estado actual de los sistemas actuales de soporte a la decisión, sistemas expertos y sistemas de minería de datos de soporte al triaje. En la sección 2 mostramos la potencialidad de las redes bayesianas en los sistemas hospitalarios de triaje, dado los buenos resultados que están teniendo para el diagnóstico de enfermedades y otras aplicaciones médicas. Exponemos asimismo los conceptos básicos relacionados con la interpretación del modelo, el aprendizaje e inferencia de las redes bayesianas, y de su uso como herramientas de clasificación. En la sección 3 se definen las redes bayesianas a usar: los distintos tipos de variables que se han tenido en cuanta y como se ha acotado el problema a un subconjunto dentro de cada grupo para restringirnos al triaje relacionado con enfermedades de origen cardiovascular y respiratorio. Asimismo se muestra la estructura de la red bayesiana creada por los expertos. Por último, los métodos de aprendizaje que serán usados para el aprendizaje automático de la estructura de la red. En la sección 4 los métodos de evaluación a utilizar que permitan comparar ambos modelos. Por último, en la sección 5 las conclusiones y futuras líneas de trabajo.

Tabla 1. Los 5 niveles de catalogación en el triaje hospitalario.

1. Resucitación	2. De emergencia	3. Urgente	4. Menos urgente	5. No urgente

2. SISTEMAS COMPUTACIONALES DE APOYO EN EL TRIAJE

Junto con la implantación de los sistemas de triaje hospitalario han aparecido diversos sistemas computacionales para el soporte a la decisión en el triaje. Entre estos se encuentran todos aquellos sistemas estáticos diseñados por médicos en los que el conocimiento se presenta por una serie de reglas que a menudo forman parte del propio código de forma que no es posible la actualización de las reglas por parte del usuario. Así ocurre por ejemplo con el "Automated Triage Management" (ATM) [12], desarrollado por expertos en medicina de urgencias de la Facultad de Medicina de la UCLA, el "Symptons, Advice, Measure" (SAM) [15], desarrollado por un médico generalista y el Ped's Advice (PA) [15], sólo para servicios de pediatría, desarrollado a nivel local por médicos y enfermeras del Hospital Académico de Niños de Upsala (Suecia). En estos dos últimos sistemas, el usuario (generalmente enfermero-as) introducen una palabra referente al mas notable de los síntomas, por ejemplo, tos y el sistema proporciona un cuestionario de respuesta si/no paso a paso.

Finalmente el sistema recomienda un nivel de triaje. Estos dos sistemas han sido criticados por sus usuarios por no ser sistemas reales de apoyo a la decisión, no estar adaptados a las situaciones de la práctica diaria en las que se hace uso de conocimiento psicológico y ético [13]. Estos programas adolecen por la falta de actualización y adaptación continua a la práctica. Cuando los programas permiten la inclusión de nuevo conocimiento por parte del usuario, generalmente en forma de reglas, deben tener claramente separado el conocimiento del código. Son los llamados Sistemas basados en el Conocimiento o Sistemas Expertos, los cuales representan el modelo de datos (como reglas o árboles de decisión) junto con los datos en la llamada Base de Conocimiento. Un ejemplo es eTRIAGE [5], el cual utiliza las reglas CTAS como base de conocimiento. eTRIAGE puede aceptar modificaciones de las reglas por parte del usuario según su juicio clínico, siempre que lo justifique. Dos ejemplos de programas cuya validez para el uso en salas de emergencias ha sido estudiado [11] son QMR (Quick Medical Reference) e Iliad. Ambos programas van más allá de la mera asignación de un nivel de triaje, pues sugieren un diagnóstico. Aunque Iliad tiene resultados significativamente mejores, ambos programas ofrecen precisiones demasiado bajas para poder usarlos como arbitraje en casos individuales. Por ejemplo, Iliad citó el diagnostico real entre los 10 más probables en un 51% de las veces (QMR el 51%) y sólo en un 36% lo citó entre los 5 más probables (QMR en un 32%). Quizás algunas de las limitaciones mas relacionadas con su falta de precisión son el hecho de que ninguno de ellos tienen en cuenta los medicamentos que los pacientes que acuden a urgencias están tomando. Tampoco permiten introducir la duración de los síntomas ni consideran la secuencia en la que los síntomas aparecen. Algunos síntomas no aparecen por ninguno de sus posibles sinónimos. Esta limitación también ocurre en SAM y PA. Estos sistemas en los cuales el conocimiento se representa en forma de reglas (sistemas basados en reglas) que pueden estar jerarquizadas (árboles de decisión) tienen en común que el modelo (las reglas o el árbol) ha sido directamente construido con el conocimiento del experto o expertos. Existen ejemplos de sistemas basados en el conocimiento en el que el conocimiento se puede inducir a partir de los datos (lo que se llama aprendizaje automático o minería de datos), en este caso síntomas, análisis clínicos y otras pruebas, signos vitales, diagnostico final, etc., como por ejemplo el Mobile Emergency Triage System (MET) [16], para triaje en niños. MET utiliza "conjuntos aproximados" (del inglés rough sets) para definir "árboles de decisión aproximados", capaces de representar incertidumbre en las reglas. Estos modelos se ajustan mejor a situaciones reales en las que existe información incompleta y asociaciones no claras entre síntomas y diagnósticos [17]. Además de los algoritmos capaces de aprender árboles de decisión, como C4.5 [20], los modelos para representar conocimiento en los que se basan otras aplicaciones que realizan aprendizaje automático pertenecen a otros muy diversos paradigmas, como por ejemplo redes neuronales, máquinas de soporte vectorial [24], algoritmos basados en instancias o redes bayesianas. Existen ejemplos de todos ellos en diagnóstico médico, como las redes neuronales para el diagnóstico de hipertensión [19], el de algoritmos basados en instancias para el diagnóstico del cáncer a partir de mamografías [2], el de las máquinas de soporte vectorial en la categorización del cáncer a partir de datos de expresión genética [25], o el de redes bayesianas para el cálculo del riesgo de un derrame cerebral a partir tanto de variables clínicas como de información genética [23]. El principal inconveniente de las redes neuronales, los algoritmos basados en instancias y las máquinas de soporte vectorial es que no aportan información legible por el hombre del modelo que relaciona las distintas variables entre si. En concreto, las redes neuronales son modelos de caja negra, y sus estructura no puede ser interpretada directamente por el experto. Esto hace a estos sistemas no útiles cuando se pretende que el experto pueda modificarlos según su conocimiento y experiencia. Los árboles de decisión y las redes bayesianas, por el contrario, tienen una interpretación directa, de forma que el modelo puede ser explicitado por un experto, o interpretado e incluso mejorado por él si se aprende de forma automática a partir de un conjunto de datos.

3 REDES BAYESIANAS

Las redes bayesianas están ampliamente aceptadas como herramientas de inteligencia artificial capaces de representar incertidumbre en la resolución de un problema [18]. Constituyen una valiosa ayuda para representar relaciones entre variables. Además permiten describir con claridad sistemas complejos mediante la información que se extrae de su estructura. La inferencia bayesiana que se realiza sobre la red facilita la comprensión de la estructura del sistema. Las relaciones dinámicas también pueden representarse, en el caso de que la relación entre variables varíe con el tiempo. Algunos ejemplos de aplicaciones recientes con buenos resultados son en su uso para la detección de correos electrónicos maliciosos [6] o en la segmentación de mercados [14].Cuando las relaciones de dependencia entre variables son causales la red se conoce con el nombre de Red Causal de Creencias (del inglés, Causal Belief Networks) o Red Probabilística Causal (Causal Probabilistic Networks).

Aunque los árboles de decisión (1) son muy útiles en la construcción de clasificadores, (2) generalmente presentan una alta fiabilidad y (3) pueden representar incertidumbre usando por ejemplo conjuntos aproximados, una jerarquía es una estructura más restrictiva de la que se necesita para representar las relaciones causa-efecto entre un conjunto de hábitos y de enfermedades (modelo predictivo) o las relaciones efecto-causa entre un conjunto de síntomas y de enfermedades (modelo de diagnostico). Las redes bayesianas constituyen una alternativa a los árboles de decisión, por permitir la representación de modelos más complejos de diagnosis o prognosis. Las redes bayesianas se basan en los fundamentos de la teoría de la probabilidad y permiten combinar el juicio del experto con las fuentes de datos disponibles, y realizar inferencia entre cualquier subconjunto de variables. En definitiva no están orientadas solo a la clasificación sino que permiten representar modelos complejos entre síntomas, diagnósticos e incluso información genética [23]. Además pueden tratar información incompleta y, en el proceso de inferencia, permiten comparar de forma sencilla sus resultados con los del experto. Aunque tienen un amplio uso en el ámbito medico, en los sistemas de triaje a penas se han empezado a aplicar. Existen algunos estudios que demuestran el mejor comportamiento de las técnicas de aprendizaje automático en la construcción de modelos de redes bayesianas que cuando son expertos médicos quienes lo hacen, como es el caso de su uso en salas de urgencia no para asignar un valor en la escala de triaje sino tan solo para la detección del síndrome coronario severo [19],[8]. La intención fue restringir el problema a un pequeño número de variables (14) que pudieran influir en la existencia del síndrome. Aunque los resultados de la red aprendida directamente eran mejores que la creada por expertos, información tal como los medicamentos que se han consumido, u otros factores sintomáticos así como la secuencia de aparición de los síntomas podrían ser de gran ayuda para mejorar la precisión de las predicciones.

Una Red Bayesiana (RB) se compone de dos partes. Por una lado, la estructura, el modelo o parte cualitativa: un grafo dirigido acíclico (GDA) donde cada nodo representa una variable aleatoria y los arcos representan dependencias probabilísticas entre las variables. Por otra parte, de una distribución condicional de probabilidades de la forma $P(x|\Pi_x)$ para cada nodo x dado su conjunto de padres $|\Pi_x$. Esta parte de la red Bayesiana se conoce como la parte paramterica o cuantitativa de la red.
La mayor asunción de independencia que se representa en una RB se llama la "Propiedad Local de Markov" [22] que consiste en que cada nodo es independiente de todos sus nodos no-descendientes dados los padres. Así, la distribución de probabilidad conjunta que representa una RB se puede obtener mediante el producto de distribuciones de probabilidades condicionales:

$$P (x_1, x_2, \ldots, x_n) = \prod_{i=1..n} P(x_i|\Pi_{xi}).$$

Para la construcción del modelo y/o los parámetros de una RB, se puede además aplicar la

estadística bayesiana que permite representar conocimiento a priori. Esto significa que la red puede tener en cuenta tanto la opinión del experto y representarla como conocimiento a priori y utilizarla para que, combinada con datos estadísticos se infiera la información a posteriori. A menudo confiar solo la opinión del experto para construir la estructura de la red no suele tener tan buenos resultados como cuando se unen datos empíricos para mejorar la precisión de la estructura. En el aprendizaje de una RB se llevan a cabo dos pasos: primero construir el modelo, después las distribuciones de probabilidad condicionada. En este trabajo, uno de los modelos construidos se ha hecho por expertos y el otro será automáticamente inferido a partir de un conjunto de datos teniendo en cuenta el conocimiento del experto como información a priori.

3.1 Clasificadores Bayesianos

Un clasificador es una función que asigna una etiqueta clase a una instancia descrita mediante un conjunto de atributos. Cuando una RB se construye con la finalidad de predecir el valor de una variable clase dada cualquier configuración en el resto de las variables, se tiene un "clasificador bayesiano". Un clasificador bayesiano que suele ser bastante preciso a pesar de su simplicidad es el "clasificador bayesiano simple" (Naive Bayes classifier) (ver Figura 1). El modelo de este clasificador asume la independencia de los atributos $\mathbf{x_1, x_2, \ldots, x_n}$ dada la clase $\mathbf{y}$. Aunque esta asunción no se cumple la mayoría de las veces, su generalmente buen comportamiento se debe a la exactitud con la que las distribuciones condicionales pueden calcularse incluso con muestras relativamente pequeñas dado que el conjunto de padres de todas las variables de entrada $\mathbf{x_1, x_2, \ldots, x_n}$ se limita tan solo a la clase: $\Pi_{x_i}=\{y\}$, ✂ i=1..n.

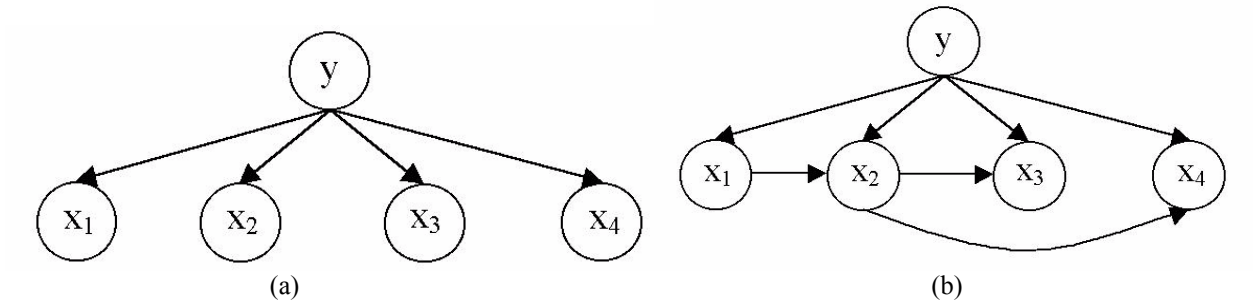


Figura 1. (a) Estructura de un clasificador bayesiano simple con 4 variables de entrada. (b) Estructura de un clasificador aumentado en árbol.

Un modelo de clasificador que mantiene un equilibrio entre simplicidad y validez es el "clasificador simple aumentado en árbol" (Augmented Naive Bayesian networks) [9], el cual permite como máximo un arco entre atributos de entrada. Para mejorar el comportamiento de un clasificador, a menudo se hace necesaria la selección automática de variables, de forma que aquellas variables que sean superfluas queden eliminadas [3].

4.UTILIZACIÓN DE REDES BAYESIANAS EN EL TRIAJE HOSPITALARIO

Un clasificador bayesiano puede definirse para el triaje hospitalario si la clase es el nivel de triaje y las variables de entradas aquella información que deba tenerse en cuanta para inferir la clase. Uno de los problemas que han sido señalados por el personal de los servicios de urgencia como inconvenientes de los sistemas computacionales de triaje es su limitación en cuanto a las variables que tienen en cuenta para determinar el nivel de triaje. Considerando esas indicaciones, las

recomendaciones de los estándares de triaje (CTAS y MAT), las realizadas por otros expertos en medicina de urgencias que colaboran en este trabajo y la tan reclamada necesidad de usar vocabularios comunes, hemos considerado el uso de 5 categorías distintas de variables: (1) síntomas (se ha usado el estándar PAT V 3.0 [10], de categorías sintomáticas, que define 32 categorías sintomáticas y 14 subcategorías que agrupan un total de 575 motivos de consulta - síntomas y síndromes - codificados según CIE-9-MC) [10], (2) el historial clínico (que puede consistir tan solo en la información que el propio paciente o sus familiares provean in situ), (3) constantes vitales, (4) variables subjetivas y (5) signos vitales.

Dada la inexistencia, en nuestro conocimiento, de sistemas de triaje basados en redes bayesianas, en este trabajo nos proponemos acotar el tipo de urgencias hospitalarias para las que el sistema debe responder. Siguiendo la opinión de los expertos hemos restringido el problema a las enfermedades cardiovasculares y respiratorias. Así, solo aquellas categorías sintomáticas que guardan más relación con estas enfermedades han sido seleccionadas como variables del grupo (1). Para las variables del grupo (2) y (4) se han tenido en cuenta también aquellas usadas por [19] para la detección del síndrome coronario severo.
El sistema que se pretende construir es un sistema en capas, de forma que existirán 3 capas determinadas por el nivel de triaje que se pretende identificar. La figura 2 presenta dicha configuración en capas. Se considera que para determinar los casos del nivel 1 no se requiere ni es práctico el uso de sistemas de apoyo a la decisión pues se debe actuar con sólo una muy rápida apreciación visual. En este trabajo nos hemos centrado en la capa externa, la más importante pues es la que pretende discernir entre el nivel de emergencia (nivel 2) y el resto de niveles menos urgentes: niveles 3, 4 y 5. En fases posteriores, se pretenden diseñar las redes bayesianas para las otras dos capas: la capa 2, a usar cuando un paciente es de nivel 3 o superior, llamada “Triaje 3” que determinara aquellos pacientes de nivel de triaje 3 frente a los de nivel 4 y 5 y la capa 4, a usar en pacientes catalogados como de nivel 4 o 5, llamada “Triaje 4” para distinguir aquellos de nivel 4 frente a los de nivel 5. Se pretende dotar también al sistema de variables explicativas o posibles diagnósticos.

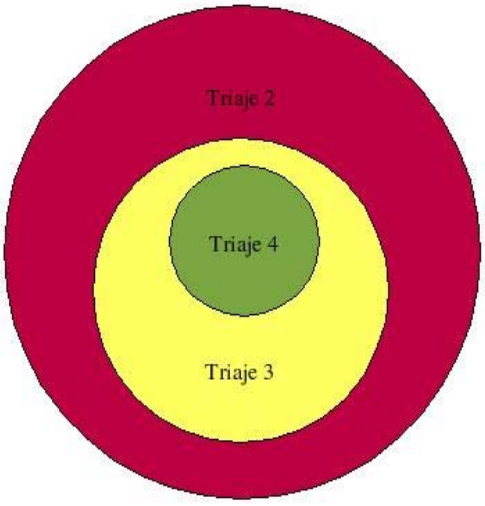


Figura 2: Arquitectura en capas del sistema de soporte a la decisión en el triaje basado en redes bayesianas.

En la tabla 2 pueden verse algunas de las variables que han sido elegidas por los expertos para la red bayesiana de la capa externa del sistema. Se pretende utilizar una muestra de más de 1000 casos para realizar le evaluación de los modelos, que será obtenida a partir de casos prácticos en salas de urgencias de hospitales de Buenos Aires. En el estudio realizado hasta el momento se ha hecho una simulación con 124 casos. Se han escogido 40 variables que han sido seleccionadas a criterio de los expertos. La tabla 2 muestra algunas de ellas, el acrónimo y la categoría a la que pertenecen. En la

figura 3 se puede ver el clasificador bayesiano aprendido a partir de los datos con el algoritmo K2 [4]. Se trata de un algoritmo para el aprendizaje automático de redes bayesianas. A partir de la red obtenida, el clasificador de la variable "triaje" puede ser obtenido con la sencilla selección de las variables que forman el manto de Markov [21] de dicha variable. Se ha utilizado el software Bayes Discoverer [22], tanto para el aprendizaje y representación del modelo como para el aprendizaje paramétrico. En un segundo paso se ha utilizado directamente un clasificador bayesiano: el clasificador bayesiano simple, sobre las mismas 40 variables. El software usado ha sido MLT [1].

Tabla 2: Algunas de las variables seleccionadas por los expertos y categoría a la que pertenecen. En total fueron seleccionadas 40 variables.

Variable	*Acrónimo*	*Categoría*
Disnea	Disnea	PAT V 3.0
Dolor torácico pleurítico	DolorTorPleu	PAT V 3.0
Ortopnea	Ortopnea	PAT V 3.0
Dolor torácico en el esfuerzo	DolorTorEsf	PAT V 3.0
Dolor torácico en reposo	DolorTorRep	PAT V 3.0
Cianosis	Cianosis	Signo vital
Intensidad del cuadro	Intensidad	Variable subjetiva
Palpitaciones	Palpitaciones	Signo vital
Aumento de mucosidad	Aumento_Muc	PAT V 3.0
Sudoración	Sudoración	Signo vital
Convulsiones	Convulsiones	PAT V 3.0
Cefaleas	Cefaleas	PAT V 3.0
Fiebre	Fiebre	Constante vital

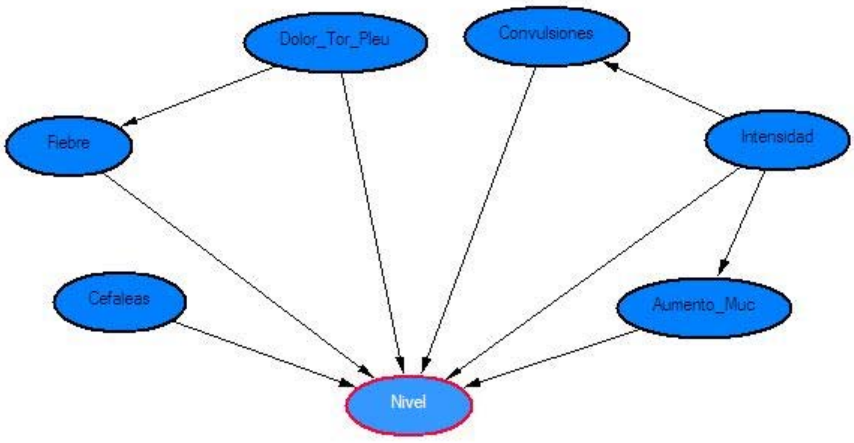


Figura 3: Estructura del clasificador bayesiano obtenido a partir del algoritmo K2.

5. EVALUACIÓN

Para analizar los datos de la simulación se han utilizado los siguientes algoritmos de aprendizaje automático:

- C4.5 [20]: Basado en la creación de un árbol de decisión, C4.5 es el más aplicado entre los de su clase (C45).
- El vecino mas cercano: Un algoritmo basado en instancias [2]. Se usarán los parámetros k=1 y k=5 (VMC1, VMC5).
- K2: Algoritmo genérico para creación de redes bayesianas.
- Clasificador bayesiano simple (Naive Bayes classifier): Especifico para clasificadores, con estimación de máxima verosimilitud (BS0) y estimador bayesiana con factor α=1 (BS1).

De cada uno de ellos se ha obtenido la precisión (en porcentaje de aciertos). Se ha utilizado validación cruzada de tamaño 5, de manera que las estimaciones mostradas son valores medios de 5 pruebas realizadas tras dividir la muestra en 5 partes y para cada prueba utilizar una parte (1/5) de los datos para hacer las estimaciones y el resto de los casos (4/5) para entrenamiento o aprendizaje del modelo. En fases posteriores se pretende también calcular la curva ROC que relaciona sensibilidad (o capacidad de detectar un caso grave cuando lo es) con especificidad (o capacidad de rechazar un caso como grave cuando es realmente no grave). La tabla 3 muestra la precisión para cada uno de los métodos empleados, ordenador por la misma:

Tabla 3. Resultados de precisión obtenidos por los 6 algoritmos empleados. Se muestran los valores medios usados en una validación cruzada de 5 hojas.

Algoritmo	*Precisión*
BS1	87,90%
K2	86,99%
BS0	86,23%
C45	79,67%
VMC5	80,63%
VNC1	80,50%

Como puede observarse, todos los algoritmos de redes bayesianas utilizados obtienen mejor precisión.

6. CONCLUSIONES Y TRABAJO FUTURO

Los árboles de decisión se vienen usando de forma convencional por representar el conocimiento de forma muy intuitiva y parecida a como los expertos lo hacen: en una serie de reglas jerarquizadas. Sin embargo, parece que en modelos relacionados con síntomas y enfermedades a menudo hay reglas inconsistentes, que han hecho que se utilicen árboles de decisión aproximados. Con este trabajo se ha mostrado el potencial que las redes bayesianas suponen para el uso en sistemas expertos en el triaje hospitalario.

En una siguiente fase de este trabajo se pretende la experimentación con casos reales. En trabajos

posteriores se puede ampliar el sistema a otras categorías sintomáticas que estén relacionadas con enfermedades de distintos orígenes. Se pretende realizar un diseño modular como se sugiere por Michalowski, et al (2003).

Agradecimientos: Este trabajo ha sido soportado en parte por el proyecto TIN2005-02516 del Ministerio de Ciencia y Tecnología de España.

8. REFERENCIAS:

[1] Abad-Grau MM, Arias-Aranda D (2006). "Operations Strategy and Flexibility: modeling with Bayesian classifiers". Journal of Industrial Management and Data Systems, 106 (4), pp. 460-484.

[2] Aha David W, Kibler Dennis, Albert Marc K (1991). "Instance-Based Learning Algorithm", Machine Learning 6, pp. 37-66.

[3] Caruana, R.A. and Freitag, D. (1994), "Greedy attribute selection", in Proceedings of the Eleventh International Conference on Machine Learning, Morgan Kaufmann Publishers, New Brunswick, NJ, pp. 28-36.

[4] Cooper GF, Herskovits E (1992). "A Bayesian method for the induction of probabilistic networks from data", Machine Learning, 9, pp. 309-347.

[5] Dong SL, Bullard MJ, Meurer DP, Blitz S, Akhmetshin E, Ohinmaa A, Holroyd BR, Rowe BH (2007). "Predictive validity of a computerized emergency triage tool". Academic Emergency Medicine 14 (1), pp. 16-21.

[6] Dong-Her, S.; Hsiu-Sen, C.; Chun-Yuan, C.; Lin, D. (2004) "Internet security: malicious e-mails detection and protection", Industrial Management and Data System, 104, pp. 613-623.

[7] Fernandes CM, Wuerz R, Cark S, Djurdjev O (1999). "How reliable is Emergency department triage?", Annals of Emergencies in Medicine, 34, pp. 141-147.

[8] Fesmire FM, Novobilski A (2004). "First step in the Erlanger Artificial Intelligence Initiative: development of a Bayesian network utilizing initial triage history to risk stratify chest pain patients for thirty-day adverse outcome.", Annals of Emergency Medicine 42, pp. 857-888.

[9] Friedman, N., Geiger, D. and Goldszmidt, M. (1997) "Bayesian network classifiers", Machine Learning, 29, pp. 131-163.

[10] Gómez Jimenez, J, Segarra X, Prat J, Fernando JB, Albert E, Borras M (2003) "Concordancia, validez y utilidad del programa informático de ayuda al triaje (PAT) del Modelo Andorrano de Triaje (MAT)". Emergencias 1 (5), pp. 339-44.

[11] Graber MA, VanScoy D (2003). "How well does decisión support software perform in the emergency department?, Emergency Medicine Journal 20, pp. 426-428.

[12] Guterman JJ, Mankovich NJ, Hiller J (2003). "Assessing the effectiveness of a computer-based decision support system for emergency department triage", Engineering in Medicine and Biology Society, 1993. Proceedings of the 15th Annual International Conference of the IEEE pp. 592 – 593.

[13] Holmstrom, Inger (2007). Decision aid software porgrams in telenursing: not used as intended?

Experiences of Swedish telenurses. Nursing and Health Siciences, 9, pp. 23-28.

[14] Kuong-Wei, W.; Kuo-Fang, P. (2002) "Market segmentation via structured click stream analysis", Industrial Management and Data System, 102, pp. 493-502.

[15] Marklund B (2000). "Symptom, Rad, Atgard". Vanersborg: Vardutveckling AB.

[16] Michalowski, Wojtek, Kersten, Wilk, Szymon, Slowinsky, Roman (2003) Designing Man-Machine interactions for Mobile Clinical Systems: MET Triage support using palm handhelds. European Journal of Operational Research, 177 (3), pp. 1409-1417.

[17] Michalowski Wojtek, Wilk Szymon, Farion Ken, Pke John et al. (2005) "Development of a decision algorithm to support emergency triage of scrotal pain and its implementation in the MET system". INFOR, Nov 2005.

[18] Nadkarni, S. and Shenoy, P.P. (2001) "A bayesian network approach to make inferences in causal maps", European Journal of Operational Research, 128, pp. 479-498.

[19] Novobilski Andrew J, Fermire Francis M, Sonnemaker David (2004). "Mining Bayesian Networks to Forecast Adverse Outcomes Related to Acute Coronary Syndrome." ." The 17th International FLAIRS Conference 2004.

[19] Poli Riccardo, Cagnoli Stefano, Livi Riccardo, Coppini Hiuseppe, Valli Guido (1991). "A Neural Network Expert System for Diagnosing and Treating Hypertension", Computer 24(3), pp. 64-71.

[20] Quinlan, JR (1996). "Improved Use of Continuous Attributes in C4.5", Journal of Artificial Intelligence Research, 4, pp. 77-90.

[21] Sebastiani P, Abad-Grau M. M., Ramoni, M. F. (2005) Bayesian Networks for Genomic Analysis, In ER Dougherty, I Shmulevich, J Chen, ZJ Wang. EURASIP Book Series on Signal Processing and Communications: Genomic Signal Processing and Statistics pp 281-320. Hindawi, New York, NY.

[22] Sebastiani P, Abad-Grau M, Ramoni MF. (2005b) Learning Bayesian networks. In Data Mining and Knowledge Discovery Handbook pp 193-230. Springer, New York, NY.

[23] Sebastiani Paola, Perls Thomas T (2007). "Complex Genetic Models", Bayesian Belief networks: a practical guide to applications. Olivier Pouret, Patrck Nam and Bruce G. Marcot (Eds.). Wiley, NY.

[24] Vapnik, V.N. (1998), Statistical Learning Theory, J. Wiley, New York.

[25] Yoonkyung Lee, Cheol-Koo Lee (2003). "Classification of multiple cancer types by multicategory support vector machines using gene expression data", Bioinformatics 19 (9), pp. 1132-1139.

Modelo de grafos para el estudio de la disponibilidad y la gestión de los Niveles de Servicio en servicios de IT

Sergio Machuca, Natalia Chiaro, Horacio Giménez y Gabriela A. Sasco

Telemática, Eduardo Acevedo 1622,
Montevideo, Uruguay, 11200
{smachuca, nchiaro, hgiménez, gsasco}@telematica.com.uy

Abstract

In the last years the organizations has increased the use of the technologies of the information, and also they increased the associated risks them.The complexity of the infrastructures that include a large number of components (applications, servers, operating systems, Databases, Application Servers, network and communications devices, UPS, power supplies, etc.) requires a careful management.
COBIT, ITIL and other proposals agree that IT Management must be aligned with services. Also agree that a configuration database (CMDB) is needed to know: components involved; dependencies between them; users and administrators; electrical and networks connexions; etc. This CMDB is the heart of the ITIL based management, because from them is possible to optimize the management of changes, incidents, problems and also the availability, capacity, continuity and service level agrees..
In this paper, we show a representation of the IT infrastructure, including its relations of dependency and network and electrical connections, etc. using ASDG, abstraction for managing changes in software components. We also show how adapt them for represent the IT component like graph nodes. Over this graph we studied the service availability with a walk around over the graph.

Keywords: Information Technology Management, Availability management, ASDG, ITIL, CMDB, impact analysis.

Resumen

En los últimos años las organizaciones aumentaron el uso de las tecnologías de la información, y también aumentaron los riesgos asociados a ellas. La complejidad de las infraestructuras que incluyen un gran número de componentes (aplicaciones, servidores, sistemas operativos, bases de datos, red y comunicaciones, UPS, generadores energía, etc.) requiere de una gestión cuidadosa.
COBIT, ITIL y otras metodologías coinciden que la gestión de IT debe estar centrada en los servicios. También coinciden que es necesaria una base de datos de configuraciones (CMDB) para conocer: los componentes involucrados; dependencias entre ellos; usuarios y administradores; conexiones eléctricas y de red, etc. Esta CMDB es el centro de la gestión basada en ITIL, pues a partir de ella es posible optimizar la gestión de los cambios, incidentes y problemas así como también de la disponibilidad, capacidad, continuidad y los acuerdos de niveles de servicio.
En este artículo presentaremos como representar la infraestructura tecnológica, sus relaciones, sus conexiones eléctricas y de red, etc., utilizando ASDG, abstracción utilizada para gestionar cambios en componentes de software. Mostramos como adaptarlo para representar los componentes de IT como nodos de un grafo, sobre el cual se estudia la disponibilidad de los servicios realizando una recorrida sobre este grafo.

Palabras claves: Information Technology Management, Configuration management, ASDG, ITIL, CMDB, análisis de impacto.

1. INTRODUCCION

En los últimos años, el uso de las tecnologías de la información en las organizaciones se ha incrementado significativamente, provocando un gran impacto en los procesos del negocio y llegando a requerir infraestructuras muy complejas. Las necesidades de negocio requieren respuestas rápidas, lo que incrementa el uso de nuevas tecnologías así como la cantidad de aplicaciones, servidores, sistemas operativos, bases de datos, servidores de aplicaciones, dispositivos de comunicaciones, generadores de energía, etc.

La complejidad de esta infraestructura se ve aumentada por los requerimientos de interoperabilidad, e-business, sistemas distribuidos, ERPs, Intranets, nuevas funcionalidades de sus sistemas legacy, accesos de diversas aplicaciones a los datos o servicios provistos por otra aplicación; y por los modelos N-TIER y más recientemente SOA.

Esta mayor dependencia de la tecnología hace que aumenten los requerimientos de disponibilidad y niveles de servicio que hacen que cualquier falla o bajada de un servicio, si no es conocida de antemano o es muy prolongada, se convierte es una situación catastrófica para la organización. Para mitigar estos riesgos, se deben tomar las medidas necesarias para asegurar que los tiempos de indisponibilidad debido a fallas sean minimizados.

COBIT [1] e ITIL [2] son quienes aportan una guía a la solución a este problema introduciendo la necesidad de la gestión de los servicios y de los procesos necesarios para mantenerlos. Es así que surgen los procesos tales como Configuration Management, Availability Management, Capacity Management, Service Level Management, IT Service Continuity Management, etc.

Estas propuestas coinciden en la necesidad de contar con una base de datos de configuraciones (CMDB) para conocer: los componentes involucrados; dependencias entre ellos; usuarios y administradores; conexiones eléctricas y de red, etc. En ITIL, esta CMDB permite una mejor gestión de los distintos procesos, optimizando la forma de gestionar los cambios, incidentes y problemas así como también la disponibilidad, capacidad, continuidad y los acuerdos de niveles de servicio.

En trabajos anteriores hemos visto que esas propuestas no incluyen la forma de modelar la infraestructura tecnológica y que existe escasa bibliografía al respecto, siendo estas en su mayoría referentes a cambios en componentes de software. En estos trabajos [3][4][5] hemos mostrado como representar la infraestructura de IT utilizando Abstract System Dependence Graph (ASDG) [6] para realizar la gestión de los cambios en los componentes de infraestructura.

En este artículo mostramos como utilizar esa representación de la infraestructura para gestionar la disponibilidad de los servicios en función de los Service Level Agreements (SLAs). El artículo se organiza como sigue: primero se realiza una introducción a los conceptos básicos de Service Level Managent, Availability Management y Configuration Management; luego se describe la forma de representar la infraestructura de IT utilizando ASDG; después vemos como utilizar esa representación para gestionar la disponibilidad de los SLA y por último mostramos nuestras conclusiones y trabajos futuros.

2. MODELOS DE GESTIÓN

Los modelos de gestión de IT más aceptados son COBIT e ITIL.

La misión definida por COBIT [1] es la de “investigar, desarrollar, publicar y promover un conjunto de objetivos de control en tecnología de información con autoridad, actualizados, de

carácter internacional y aceptados generalmente para el uso cotidiano de gerentes de empresas y auditores".

COBIT hace énfasis en la orientación al negocio y ha sido diseñado no solo para ser utilizado por usuarios y auditores, sino que en forma más importante, está diseñado para ser utilizado como una lista de verificación detallada para los propietarios de los procesos de negocio. En el Marco Referencial de COBIT se proporcionan herramientas al propietario de procesos de negocio que facilitan el cumplimiento de sus responsabilidades. El marco referencial comienza con la premisa "Con el fin de proporcionar la información que la empresa necesita para alcanzar sus objetivos, los recursos de TI deben ser administrados por un conjunto de procesos de TI agrupados en forma natural.

COBIT define un conjunto de 34 objetivos de control de alto nivel, uno para cada uno de los procesos de TI, agrupados en cuatro dominios: planeación & organización, adquisición & implementación, entrega (de servicio) y monitoreo. En particular se pueden mencionar los siguientes: DS1 Definir Niveles de Servicio, DS3 Administrar Desempeño y Capacidad, DS4, Asegurar Servicio Continuo, DS6 Identificar y Asignar Costos, DS8 Apoyar y asistir a los Clientes de TI, DS9 Administrar la Configuración, DS10 Administrar Problemas e Incidentes.

Por su parte ITIL [2][7][8][9][10][11] fue creada para comunicar las mejores prácticas en la Gestión de Servicios TI. ITIL ofrece un marco para todas las actividades de TI, como parte de la entrega de servicios, basado en la infraestructura TI, Las actividades se dividen en procesos, que proporcionan el marco para gestionar los Servicios TI en forma más madura.

ITIL provee de "mejores prácticas" para la gestión de TI las cuales proveen:

- Guías para alinear los servicios de TI con los requerimientos de negocio
- Un lenguaje común para TI y el negocio
- Un marco referencial, no una metodología
- Un conjunto de mejores prácticas neutral a los proveedores
- Guías, no un como hacerlo paso a paso para que una organización implemente procesos de gerenciamiento de TI

- Además del modelo de procesos se encuentran:
  - Guías en la planificación e implementación
  - Sugerencias de organización, roles y habilidades requeridas
  - Sugerencias para la educación y el entrenamiento
  - Descripción de atributos clave en herramientas
  - Ejemplos de políticas y procedimientos

La parte más importante de ITIL es el "Service Management" el cual se compone de dos libros "Soporte del Servicio" (Service Support) y "Entrega del Servicio" (Service Delivery).

En soporte del servicio se tratan los siguientes procesos

- Relaciones entre procesos
- La función Service Desk
- Gestión de Incidentes
- Gestión de Problemas
- Gestión de Configuraciones
- Gestión de Cambios
- Gestión de Versiones
- Herramientas de Gestión de Servicios

En Entrega del servicio se tratan los siguientes procesos

- Relación entre procesos
- Gestión de Niveles de Servicio
- Gestión Financiera de los Servicios de TI
- Gestión de Capacidad
- Gestión de Continuidad de los servicios de TI
- Gestión de Disponibilidad
- Herramientas de Gestión de Servicios

Es posible decir que la Gestión de los Niveles de servicio es la base del planteo de ITIL. Al analizar sus procesos, podemos ver que la Gestión de las configuraciones es la que brinda, a través de su CDMB (Configuration Management Data Base) la información necesaria para la mayoría de los procesos, pues es la que permite conocer la infraestructura propiamente dicha.

MOF [12] es una propuesta de Microsoft basada en ITIL que, al igual que otras utilizadas por empresas como HP e IBM, podemos considerar que comparte esta visión. Lo mismo puede decirse de otras propuestas [13].

A grandes rasgos podemos ver que COBIT e ITIL persiguen objetivos similares. Ambas propuestas plantean la necesidad de identificar, controlar y mantener todos los componentes de TI, de gestionar los niveles de servicio, la disponibilidad, la capacidad, etc.

3. METODOLOGÍA PROPUESTA

En [3][4][5] hemos visto una forma de representar la infraestructura de TI utilizando ASDG. En esos trabajos, vimos como es posible realizar estudios de impacto previos a la realización de un cambio de forma de disminuir los riesgos ocasionadas por cambios en la infraestructura. En este artículo mostraremos como usar esa representación para mejorar la gestión de los niveles de servicio y en particular como calcular la disponibilidad de los servicios.

3.1 Especificación del problema

Utilizaremos Abstract System Dependence Graph (ASDG) [6][14][15], abstracción de grafos de dependencia, para representar toda la infraestructura de TI como un grafo, donde los servidores, dispositivos de red, software de base, aplicaciones, dispositivos eléctricos y usuarios, son nodos y sus relaciones de dependencia, conexiones físicas y eléctricas, son aristas.

Primero veamos una definición de ASDG y a luego como utilizarlo para resolver los problemas planteados.

3.2 Abstract System Dependence Graph

Un grafo de dependencia de sistema (SDG) [14][15] se compone de varios grafos de dependencia de procedimientos. En [6] se propone un grafo abstracto de dependencia de sistema (ASDG) que puede ser construido usando un subconjunto de información del SDG. El hecho que ASDG es un subconjunto de SDG garantiza que los algoritmos usados para construir SDG pueden usarse también para construir el ASDG.

ASDG consiste en vértices que representan componentes, por ejemplo funciones y variables globales. La función call es representada por la arista call, y la arista data flow representa un flujo de datos de una función a una variable global y viceversa.

Formalmente, se define ASDG de la siguiente manera:

Sea C el conjunto de componentes en el software, y $C = F \cup G$

donde **F** es un conjunto de funciones y

G es un conjunto de datos globales

Para d,e ∈ C, la arista <d,e> denota una dependencia de componente d en e.

Si d, e ∈ F entonces <d,e> es la arista *call*, si d ∈ G o e ∈ G, entonces la arista <d,e> son los flujos de datos.

Un ASDG es un conjunto de dependencias D en el sistema.

El conjunto de todos los componentes usado en D se define como

comp(D) = { e | si existe d, tal que <e,d> ∈ D o <d,e> ∈ D}

Los vecinos de un vértice d se definen como

neigh(d) = { <e,d > | si existe e tal que <e,d > ∈ D} ∪

{ <d,f> | si existe f tal que <d,f> ∈ D}.

Cabe aclarar que se usa la notación de [2, 3] para los grafos.

Un ejemplo de ASDG y su código fuente se puede apreciar en Figura 1.

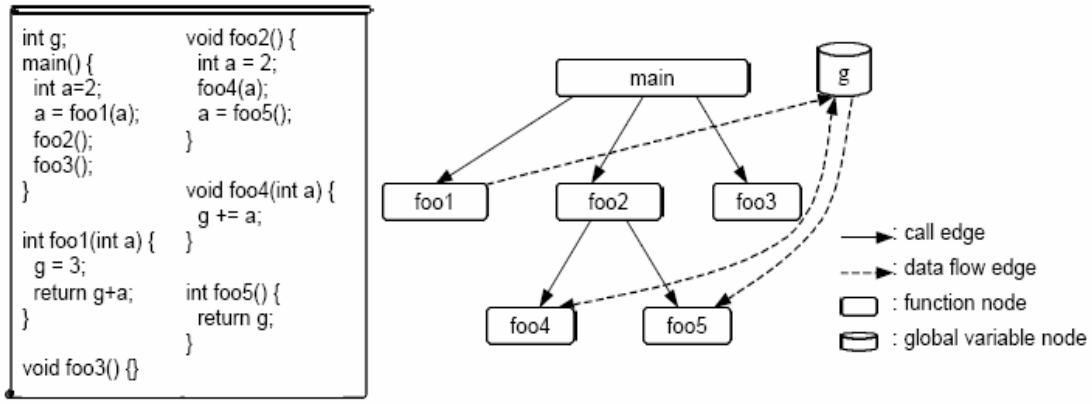


Figura 1 – Un ejemplo de programa y su ASDG

3.3 Representación de la infraestructura de TI

Ahora mostraremos como adaptar ASDG, para representar nuestro problema.

Como hemos visto anteriormente, necesitamos representar los componentes de infraestructura de nuestra red, sus conexiones y dependencias.

Para esto, consideremos los componentes como nodos de nuestro grafo de dependencias y las dependencias como aristas del mismo.

Por consiguiente, definamos formalmente el grafo G de la siguiente manera:

$$G = C \cup RRHH \cup R$$

donde

C son los componentes de la infraestructura
RRHH son recursos humanos de la organización
R son las relaciones entre los componentes

A continuación se detallarán los mismos.

Sea **C (componentes de la infraestructura) = F ∪ L ∪ E ∪ O**

F es el conjunto de componentes físicos de la infraestructura
L es el conjunto de componentes lógicos de la infraestructura
E es el conjunto de componentes eléctricos de la infraestructura
O es el conjunto de otros componentes de la infraestructura

F (componentes físicos) = N ∪ H ∪ OTF

N = HUB ∪ SWT ∪ ROU ∪ FRW
HUB conjunto que representa los hub de la infraestructura
SWT conjunto que representa los switch de la infraestructura
ROU conjunto que representa los router de la infraestructura y
FRW conjunto que representa los firewall de la infraestructura

H = EQP ∪ P, con P = IMP ∪ SCN ∪ MON
EQP es el conjunto que representa los equipos de la infraestructura
IMP es el conjunto que representa las impresoras de la infraestructura
SCN es el conjunto que representa los scanner de la infraestructura y
MON es el conjunto que representa los monitores de la infraestructura
OTF el conjunto que representa otros componentes físicos

L (componentes lógicos) = SO ∪ SRV ∪ OTL
SO el conjunto que representa los sistemas operativos
SRV el conjunto que representa los servicios
OTL el conjunto de representa otros componentes lógicos

E (componentes eléctricos) = UPS ∪ GEN ∪ ENC ∪ CDD ∪ OTE
UPS el conjunto que representa las UPS
GEN el conjunto que representa los generadores
ENC el conjunto que representa los enchufes
CDD el conjunto que representa las cajas de distribución
OTE el conjunto que representa otros componentes eléctricos

Los **recursos humanos** se definen de la siguiente manera: **RRHH = ADM ∪ DES ∪ USU**
ADM el conjunto que representa los administradores de recursos
DES el conjunto que representa los desarrolladores
USU el conjunto que representa los usuarios de un servicio

Sea **R** el conjunto de relaciones del sistema,

$$\mathbf{R = RC \cup RD}$$

Donde:
RC es el conjunto de aristas que representan relaciones de conexiones de red
RD es el conjunto de aristas que representan relaciones de dependencia de la infraestructura

La arista <d,e> ∈ RD denota una dependencia funcional del componente d en e.

RD = {<d,e> | si y solo si se cumple que existe una dependencia funcional del componente d en e}

En [4] vimos que la inclusión de las RC nos obliga a extender al ASDG con un nuevo tipo de arista, por lo que para las conexiones de red definimos un grafo estándar en el cual los nodos son los componentes del ASDG y las aristas son las propias conexiones.

Ahora ya tenemos definida completamente la infraestructura tecnológica incluyendo a los componentes de hardware, software, sus administradores y los usuarios de los servicios.

3.4 Análisis

3.4.1 Análisis de impacto

Cuando es necesario realizar un cambio en cualquiera de los componentes, se desea conocer todos los componentes y recursos humanos (administradores, usuarios) impactados. Para ello se construye un árbol de alcance a partir del nodo que representa el componente, que incluya todos los nodos alcanzados por las relaciones de dependencia con el mismo. El árbol incluirá entonces todos los componentes impactados, los administradores y los usuarios.

En el caso en que se impacte un componente de red, el algoritmo presenta algunas variaciones. En este caso se debe verificar que para todo par de componentes, que residen en equipos distintos y que existe una dependencia entre ellos, exista un camino de red (que no pase por los componentes de red afectados) entre los componentes de hardware donde reside cada uno. En caso de no existir se debe incluir en el árbol el componente dependiente en la relación y continuar con el árbol de alcance.

3.4.2 Análisis de componentes de un servicio

Podemos ver que esta estructura que hemos construido, contiene todos los servicios que se ofrecen y de que componentes dependen.

Conocer todos los componentes involucrados en un servicio nos ayudaría a planificar la calidad de dichos servicios.

En nuestra representación, los servicios se corresponden con los usuarios del mismo.

Si realizamos una recorrida en dirección contraria en el grafo, o sea que comenzamos desde un *usuario de un servicio* y vamos construyendo un árbol de alcance, con los componentes de quien se depende, tendremos la lista de todos los componentes involucrados en la prestación del servicio. Esta lista nos permite, estudiar los posibles cuellos de botella e invertir en la mejora de los puntos que puedan ser críticos para optimizar la disponibilidad, la capacidad y la continuidad de los servicios.

3.4.3 Disponibilidad de los servicios

Tradicionalmente, los estudios de disponibilidad estaban centrados en los distintos componentes, antes que en los servicios en si. Las propuestas de COBIT e ITIL nos expresan la necesidad de conocer la disponibilidad de los servicios. En nuestra representación, podemos observar que la falla de cualquier componente afecta la disponibilidad de distintos servicios, por lo que para un servicio, su tiempo de indisponibilidad, es la sumatoria de las indisponibilidades de todos los componentes involucrados en la prestación de dicho servicio.

Para calcular la disponibilidad de los servicios, se debe ir contabilizando las indisponibilidades ocasionadas por fallas en cada componente.

Ante la indisponibilidad de un componente, se realiza el análisis de impacto descrito en 3.4.1 y del árbol de alcance se extraen los servicios impactados. Para cada servicio impactado se incrementa el tiempo de indisponibilidad.

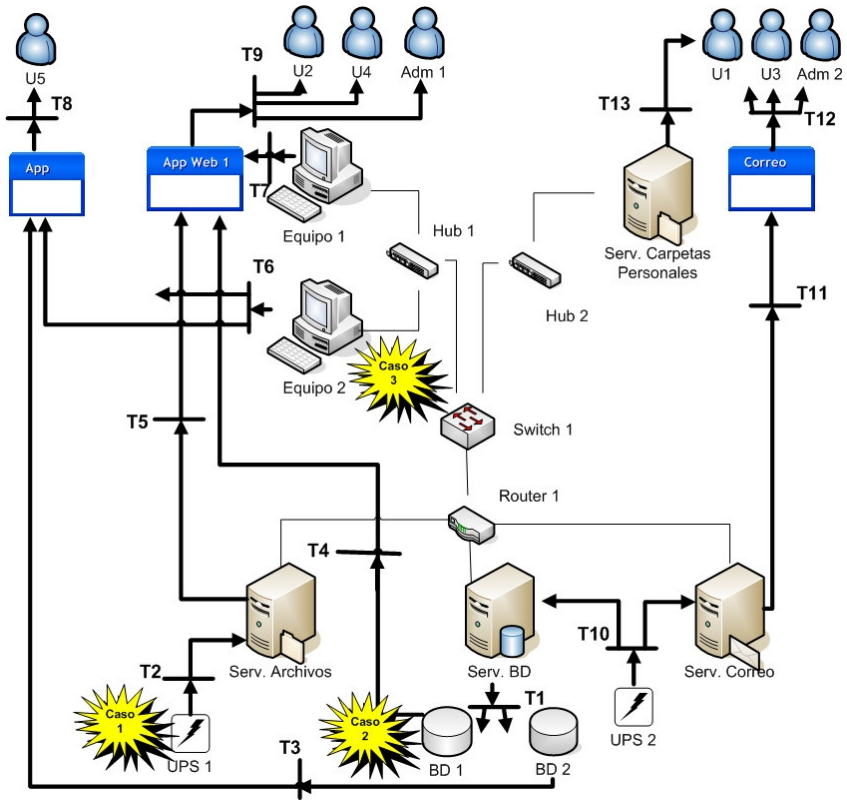
Finalmente es posible calcular la disponibilidad

$$\textbf{\% Disponibilidad} = \frac{\text{Horas Acordadas de Servicio} - \sum(\text{indisponibilidad componente i})}{\text{Horas Acordadas de Servicio}} \times 100$$

3.5 Caso de Estudio

Consideremos ahora la siguiente realidad representada gráficamente en la siguiente figura.

En ella se pueden apreciar los componentes de red, como por ejemplo Hub 1, Hub 2, Switch 2, Router 1, además de los componentes de hardware como los equipos, los servidores, etc. También se pueden observar las distintas dependencias que hay entre los mismos (líneas punteadas) y los usuarios que dependen de ellas..



3.5.1 Análisis de impacto

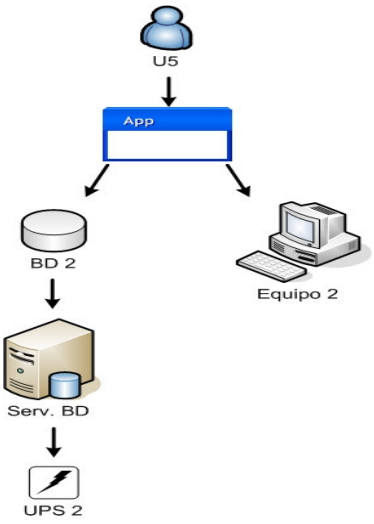
En [3][4][5]hemos mostrado este análisis. El árbol de alcance que se obtiene se utiliza por el proceso de Gestión de Cambios para la planificación y autorización de los cambios.

3.5.2 Análisis de componentes de un servicio

El servicio ofrecido por la aplicación ***APP*** es utilizado por el usuario ***U5***. Como ya comentamos en nuestra representación, existe una correspondencia entre *usuario* y *servicio*, por lo que el estudio lo realizaremos a partir del usuario ***U5***.

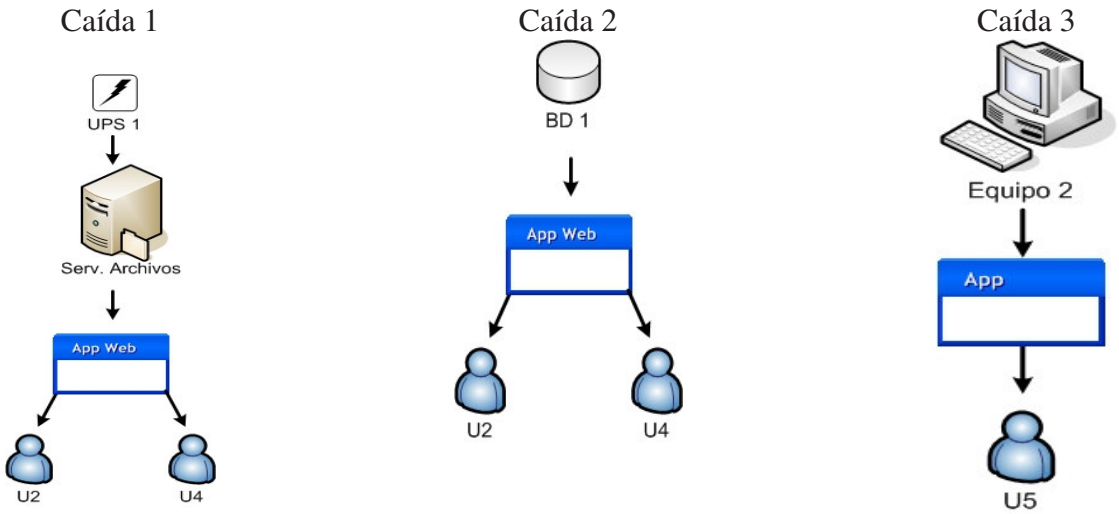
En la siguiente figura, podemos observar, que el servicio utilizado por ***U5*** depende de ***APP***, quien a su vez depende de ***BD2*** y ***Equipo2***. Siguiendo con el análisis se observa que ***BD2*** depende de ***ServidorBD*** y de ***UPS2.***

Esta información, al momento de definir los alcances del servicio, nos permitirá evaluar las posibilidades y ajustar las necesidades de mejora o en su defecto limitar los alcances.



3.5.3 Cálculo de disponibilidades

A continuación analizaremos la disponibilidad de los distintos servicios, teniendo en cuenta las siguientes caídas.



Teniendo en cuenta las caídas anteriores, veamos a continuación, la disponibilidad de los distintos servicios: App (usuario U5), AppWeb (usuarios U2 y U4) y Correo (usuarios U1 y U3)

Disponibilidad servicio APP = $\frac{\text{Hs Ac Serv} - (\text{indisp caída 3})}{\text{Hs Ac Ser}} \times 100$

Disponibilidad AppWeb = $\frac{\text{Hs Ac Serv} - (\text{Hs indisp caída 1} + \text{Hs indisp caida 3})}{\text{Hs Ac Serv}} \times 100$

Disponibilidad correo = $\frac{\text{Hs Ac Serv} - (0)}{\text{Hs Ac Serv}} \times 100 = 100\%$

Podemos observar que el servicio de correo no fue afectado por ninguna de las caídas, por lo que su disponibilidad es del 100%. Los servicios brindados por APP y por AppWeb fueron afectados por caídas, por lo que el cálculo de disponibilidad tiene en cuenta las mismas.

4. CONCLUSIÓN Y TRABAJOS FUTUROS

Hemos analizado la problemática de la Gestión de los servicios de IT y en particular la necesidad de contar con una CMDB como soporte de diversos procesos de la gestión. Esta CMDB nos permite entre otras cosas realizar un análisis de impacto previo a la implementación de un cambio de configuraciones en la infraestructura tecnológica de una organización, conocer todos los componentes involucrados en un servicio, calcular la disponibilidad asociada a cada servicio en función de los SLA.

Hemos visto que este análisis puede ser realizado visualizando los componentes de la infraestructura como un grafo sobre el cual, realizando recorridas, se pueden efectuar análisis de

alcance y con ello identificar los componentes involucrados. Hemos usado ASDG, abstracción basada en grafos utilizada para el manejo de cambios en componentes de software, para analizar los cambios de la infraestructura.

Hemos visto que ASDG no nos permite una representación completa de la infraestructura y hemos mostrado como resolverlo. Además, el modelo no incluye algunos puntos importantes como ser los las configuraciones redundantes de componentes, configuraciones de clusters, alta disponibilidad, balanceo de carga y contingencia y también el factor tiempo (es decir, que una dependencia es válida en determinada franja horaria).

Nuestros planes incluyen tratar con los puntos mencionados en el párrafo anterior, así como también trabajar en la optimización de los algoritmos utilizados y estudiar algunas alternativas de representación, entre las que se encuentra la utilización de Redes de Petri.

REFERENCES

[1] COBIT. http://www.isaca.org/, Ult. acc. 2007.

[2] ITIL. http://www.itsm.org/, http://www.itil.org, http://www.itilsurvival.com, Ult. acc. 2007.

[3] S.Machuca, G.Sasco, N.Chiaro: Un modelo basado en grafos para análisis de impacto en cambios de componentes de infraestructura, JAIIO-ASIS 2005.

[4] S.Machuca, G.Sasco, N.Chiaro: Análisis de impacto en la gestión de cambios de configuración de componentes de IT, CLAIO (Conferencia Latinoamericana de Investigación Operativa) 2006.

[5] S.Machuca, G.Sasco, N.Chiaro: Análisis de impacto en la gestión de cambios de servicios de Telecomunicaciones, MVD TELCOM 2006 (I Congreso Regional de Telecomunicaciones).

[6] Kunrong Chen, Václav Rajlich, Case Study of Feature Location Using Dependence Graph.. Department of Computer Science, Wayne State University, 2002, pp. 293 - 299.

[7] Charles Thomas Betz, The convergence of metadata and IT service management, 2003.

[8] M. Berkhout, R. Harrow, Service Support: Service Desk and the Process of Incident Management, Problem Management, Configuration Management, Change Management and Release Management, London, The Stationery Office, 2000.

[9] A. Cassidy, K. Guggenberger A Practical Guide to Information Systems Process Improvement. Boca Raton, FL, St. Lucie, 2001.

[10] IT Service Management Forum, IT Service Managment Forum, USA, Attendee FAQ, 2003.

[11] J. Van Bon, G. Kemmerling, IT service management : an introduction, Canada, 2002.

[12] MOF. http://www.microsoft.com/technet/itsolutions/cits/mo/mof/default.mspx, Ult. acc. 2006.

[13] Erp4it: Managing information systems, http://erp4it.typepad.com/erp4it/2004/10/cmdb_chaos_and_.html, Ult. acc. 2006.

[14] Horwitz, S.; Reps, T and Binkley, D. Interprocedural Slicing Using Dependence Graphs ACM Trans. Programming Languages and Systems, vol 12, No. 1 (Jan 1990), pp. 26-60.

[15] Horwitz, S.; Reps, T. The Use of Program Dependence Graphs in Software Engineering. Proceedings of the 14th International Conference on Software Engineering, (May 1992).

Paralelización de los Procesos de Búsqueda y Optimización en Memoria Secundaria para la Estructura *Spaghettis*

Carlos Subiabre[1], Enrique Árias[2], and
Roberto Uribe-Paredes[1,3]

[1] Depto. de Ingeniería en Computación*,
Universidad de Magallanes, Chile
[2] Departamento de Sistemas Informáticos,
Escuela Politécnica Superior de Albacete, Universidad de Castilla-La Mancha, Albacete, España
[3] Grupo de Bases de Datos - UART,
Universidad Nacional de la Patagonia Austral, Rio Turbio, Argentina
E-mail: (casubiar@ona.fi.umag.cl, earias@dsi.uclm.es, ruribe@ona.fi.umag.cl)

Abstract Searching by similarity in metric spaces allows to retrieve similar or alike object to a given query. Some applications, require that the search process has to be done in real-time.
Thus, in the present work shows a parallel implementation of the similarity search by using the *Spaghettis* data structure but it is enhanced for secondary memory. The parallel implementation is based on a coarse grain approach by using *MPI*. The experimental results show good performances of the implementation in terms not only on distance evaluation but also in terms on execution time, speed-up and efficiency.
Keywords: Databases, data structures and algorithms, similarity search, parallel processing, MPI.

Resumen La búsqueda por similaridad en espacios métricos permite recuperar obejstos similares o parecidos a una consulta dada. Algunas aplicaciones, requieren que dicha búsqueda se realice en tiempo real.
Por tanto, en el presente trabajo muestra una implementación paralela del algoritmos de búsqueda por similaridad utilizando la estructura de datos *Spaghettis* pero optimizada para memoria secundaria. La implementación paralela está basada en una aproximación de grano grueso utilizando *MPI*. Los resultados experimentales muestran buenas prestaciones de la implementación en términos no sólo de evaluaciones de distancia sino también en términos de tiempo de ejecución, speed-up y eficiencia.
Keywords: Estructuras de datos y algoritmos, bases de datos, búsqueda por similaridad, paralelismo, MPI.

1. Introducción

1.1. Antecedentes

Con la rápida evolución de las tecnologías de la información han surgido nuevos depósitos no estructurados de datos tales como texto libre, imagen, sonido y video. Realizar búsquedas exactas sobre estos datos sería poco útil. Por ejemplo, si se consultase por un elemento sobre una base de datos de imágenes, la consulta sólo podría encontrar su copia digital exacta en la base de datos. El verdadero interés reside, por ejemplo, en consultar sobre una base de datos de fotografías una imagen que contiene un rostro, donde no necesariamente existe una copia exacta de la misma fotografía; identificación de individuos a través de dispositivos biométricos, donde el dato consulta (voz, retina, etc) podría verse afectado por factores externos; encontrar una especie más parecida a otra en una base de datos de cadenas de ADN, etc. Este tipo de búsqueda recibe el nombre de *búsqueda por similaridad* y consiste en recuperar todos los objetos

* Parcialmente financiado por proyecto Fondecyt 1060776, Conicyt y programa de investigación PR-F1-002IC-06, Universidad de Magallanes, Chile.

mas relevantes o parecidos a una consulta dada.
Para manipular dichos datos, se deben generar estructuras que permitan almacenarlos y realizar búsquedas sobre ellos. Estructurar este tipo de datos es dificultoso ya sea manual o computacionalmente y restringe de antemano los tipos de búsqueda posibles.
En la actualidad, la mayoria de las estructuras han sido diseñadas como prototipos y carecen de dos características importantes, dinamismo y un adecuado desempeño en memoria secundaria. La segunda, que es la más escasa, determina el desempeño en términos de accesos a disco y espacio utilizado por la estructura. Finalmente, la necesidad de procesar grandes volumenes de datos obligan a aumentar la capacidad de procesamiento y con ello la paralelización de algoritmos y distribución de la base de datos.

1.2. Marco Teórico

La similaridad, en muchos casos, es modelada a través de un *espacio métrico* y la búsqueda de objetos más similares bajo una función conveniente de similaridad, a través de una búsqueda por rango o vecinos más cercanos.

Definición 1 (*Espacios Métricos*): Un espacio métrico es un conjunto X con una función de distancia $d : X^2 \rightarrow R$, tal que $\forall x, y, z \in X$,

1. $d(x, y) \geq 0$ *and* $d(x, y) = 0$ *ssi* $x = y$. *(positividad)*
2. $d(x, y) = d(y, x)$. *(Simetría)*
3. $d(x, y) + d(y, z) \geq (d(x, z)$. *(Desigualdad Triangular)*

Definición 2 (*Consulta por Rango*): Sea un espacio métrico (X,d), un conjunto de datos finito $Y \subseteq X$, una consulta $x \in X$, y un rango $r \in R$. La consulta de rango alrededor de x con rango r es el conjunto de puntos $y \in Y$, tal que $d(x, y) \leq r$.

El objetivo de los algoritmos de búsqueda es minimizar la cantidad de evaluaciones de distancia realizadas al resolver la consulta. Los métodos para buscar en espacios métricos se basan principalmente en dividir el espacio empleando la distancia a uno o más objetos seleccionados.

Existen dos métodos para las construcción de estructuras métricas, los basados en Clustering y los basados en Pivotes. El primero divide el área en particiones de Voronoi, donde existe un centro por cada área y los demás objetos se almacenan en el centro más cercano.
En el caso de los *Algoritmos Basados en Pivotes*, un pivote es un objeto preseleccionado y que no necesariamente pertenece a la base de datos. Su objetivo es filtrar objetos en una consulta a través de la utilización de la desigualdad triangular, sin medir realmente la distancia entre el objeto consulta y los objetos descartados.

Una visión abstracta de los algoritmos basados en pivotes es la siguiente:

- Se selecciona un conjunto de k pivotes $\{p_1, p_2, \ldots, p_k\} \in X$. En tiempo de indexamiento, para cada objeto x de la base de datos Y se calcula y almacena su distancia a los k pivotes $(d(x, p_1), \ldots, d(x, p_k))$.
- Dada una consulta (q, r), por desigualdad triangular se tiene que $d(p_i, x) \leq d(p_i, q) + d(q, x)$, con $x \in X$, de la misma forma se tiene que $d(p_i, q) \leq d(p_i, x) + d(q, x)$. De estas inecuaciones se tiene que una cota inferior para la distancia entre q y x es $d(q, x) \geq |d(p_i, x) - d(p_i q, q)|$. Como los objetos x que intersectan son aquellos en donde $d(q, x) \leq r$, entonces se pueden excluir todos los objetos que no cumplan con la condición de la ecuación (1).

$$|d(q,p_i) - d(x,p_i)| \leq r, \forall i = i...k \quad (1)$$

En otras palabras, si para algún pivote p_i se cumple que $|d(q,p_i) - d(x,p_i)| > r$, entonces por desigualdad triangular se conoce que $d(q,x) > r$ y por lo tanto, no es necesario evaluar explicitamente $d(x,q)$. Todos los objetos que no se puedan descartar por esta regla deben ser comparados directamente con la consulta.

Existen distintas estructuras para buscar en espacios métricos, las cuales pueden ocupar funciones discretas o continuas de distancia. Algunos son BKTree [3], MetricTree [10], GNAT [2], VpTree [13], FQTree [1], MTree [5], SAT [8,9], EGNAT [11].

El presente artículo desarrolla una paralelización de los algoritmos de búsqueda permitiendo una una mayor eficiencia dada una consulta por rango. La estructura fue acondicionada para un buen desempeño en memoria secundaria en términos de accesos a disco. Para la implementación paralela se utilizó la librería llamada MPI (Message-Passing Interface), la cual proporciona independencia de la arquitectura y ha demostrado eficiencia en aplicaciones tales como texto y otros.

2. *Spaghettis* en Memoria Secundaria

La estructura *spaghettis* [4] es una variante de *LAESA* [7]. La Estructura aquí presentada varía de su algoritmo original ya que está optimizada para memoria secundaria [12]. Es una estructura del tipo arreglo y su algoritmo está basado en pivotes.

2.1. Construcción de Multiples *Spaghettis*

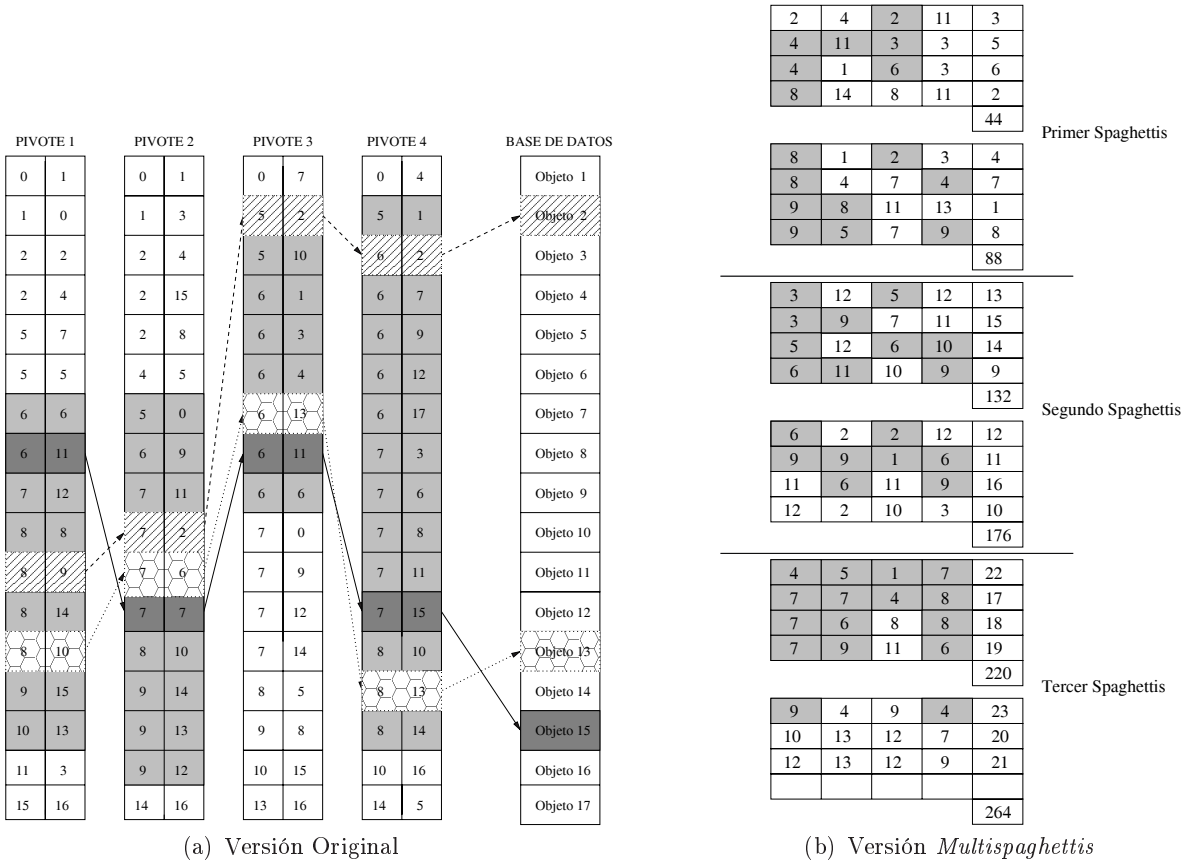
- Se selecciona un conjunto de k pivotes.
- Se calcula la distancia de cada objeto a todos los pivotes y se insertan en forma ordenada en los nodos, este orden se hace en base a la distancia del objeto con el primer pivote. Cada nodo corrresponderá a una página de disco.
- Al momento de tener la cantidad máxima de nodos soportados en RAM y estos estén totalmente llenos se procede a su almacenamiento en memoria secundaria, posteriormente se continúa con la construcción de un nuevo *spaghettis*.
- Al momento de ir guardando los nodos, uno a uno, se utiliza una estructura que almacena la ubicación del primer nodo del conjunto almacenado, de forma de poder determinar donde se encuentra ubicado dentro del índice el subconjunto de nodos formados.

En resumen, la diferencia entre la versión original de la estructura y ésta, es que se tienen N *spaghettis* o *subspaghettis*, con capacidad para *max* nodos, que corresponden al número de nodos que soporta la RAM.

2.2. Búsqueda

- Se determina la distancia entre la consulta q y cada pivote p_k, obteniendo así un intervalo por cada pivote de la forma:
$$[d(q,p_k) - r, d(q,p_k) + r] \quad (2)$$
- Se carga en RAM nodo a nodo hasta encontrar, un valor de distancia al primer pivote, que satisfaga el intervalo (2) para éste pivote.
- Un objeto candidato a la consulta es aquel que se encuentra en la intersección de los k intervalos formados.

- Para cada objeto candidato obtenido, se calcula la distancia entre él y la consulta q. Si la distancia obtenida está dentro del rango r, entonces el objeto es respuesta a la consulta realizada. Es decir, se cumple que $d(x, q) \leq r$.
- Lo anterior se realiza para cada *subspaghettis*.



(a) Versión Original (b) Versión *Multispaghettis*

Figura 1. Representación de dos versiones de *spaghettis* una vez finalizado el proceso de construcción y mostrando los intervalos obtenidos para distintas búsquedas.

Para notar la diferencia entre ambas estructuras (ver figura 1), cabe mencionar que en la estructura original las columnas de distancias están ordenados según la distancia a cada pivote, en la nueva versión sólo están ordenados según el primer pivote y por *subspaghetti*.

En la figura 1(a) se representa la forma original de la estructura *spaghettis* (solo para RAM). Ésta está construida usando 4 pivotes para indexar una base de datos de 17 objetos. Sobre esta estructura se realiza la búsqueda como sigue. Suponga una consulta q con distancia a los pivotes {8,7,4,6} y rango de búsqueda r=2 . En la figura 1(a) se muestran más oscurecidos los intervalos {(6,10),(5,9),(2,4),(4,8)} sobre los cuales se realizará la búsqueda. En la misma figura se aprecian con distintos achurados todos los objetos que pertenecen a la intersección de todos los intervalos. Dichos objetos son posibles candidatos a ser solución.

La opción implementada en este trabajo es la explicada anteriormente (*multispaghettis*). La figura 1(b) representa esta estructura una vez finalizada la construcción, es decir, luego del ordenamiento de los objetos en base al primer pivote (en el ejemplo, usando 23 objetos). En esta implementación cada fila representa a un objeto y cada columna de cada objeto corresponde

a la distancia al pivote. La ultima columna representa el identificador del objeto. En el ejemplo se asume que una página de disco mide 44 bytes y que la RAM sólo tiene capacidad para dos nodos, por ello se tienen 3 spaghettis. En la figura 1(b), se muestra también un ejemplo de búsqueda para una consulta q con $d(q, p_i) = \{6, 8, 3, 7\}$ y $r = 3$. Se muestran oscurecidas todas las celdas de todos los pivotes que están dentro del intervalo. Sin embargo, en términos reales se descarta inicialmente en base al primer pivote.

3. Implementación Paralela de los Procesos de Búsqueda de la Estructura Spaghettis

El objetivo principal para el presente trabajo, es reducir los *tiempos de ejecución* y *evaluaciones de distancia* que se realizan durante la búsqueda, así como obtener para que pivote o pivotes la búsqueda se comporta más eficiente. Esto, dado la necesidad de respuestas rápidas en diferentes aplicaciones de tipo real, como por ejemplo, durante la identificación de individuos mediante dispositivos biométricos.

Se considera que el sistema trabaja bajo régimen estacionario, por lo que la fase de construcciñon no ha sido considerada en este trabajo para ser paralelizado. Los resultados experimentales mostrados en el artículo corresponden a una estructura construida sobre un computador maestro (o broker) y replicada a todos los procesadores. Las consultas fueron distribuidas en porcentajes similares sobre el clúster de PCs, procesadas localmente y las respuestas recuperadas fueron recopiladas por el maestro.

Para los experimentos se utilizó un espacio de palabras en español de 86.061 objetos, de donde el 90 % se utilizó para construir la estructura y el restante 10 % corresponde al conjunto de consultas que se distribuyen en el cluster.

Para medir la distancia en un espacio de palabras, se utilizó la *distancia de edición* o *distancia de Levenshtein*. La distancia de edición se define como la cantidad de inserciones, eliminaciones o modificaciones de caracteres que deben realizarse sobre una palabra para convertirla en otra.

Las pruebas realizadas se efectuaron construyendo la estructura para 4, 8, 12, 16 y 20 pivotes. La selección de pivotes es aleatoria y son extraídos desde la base de datos. El tipo de búsquedas fue para *rangos* 1, 2, 3 y 4.

Los experimentos fueron ejecutados sobre un clúster compuesto por 9 procesadores (un front-end y 8 procesadores para cómputo). Cada procesador esta compuesto por un Procesador Intel PIV de 3.00 GHz y 3GB de RAM. Este corre sobre el sistema operativo Debian Linux. Para realizar la paralelización se utilizó MPI [6] (Interfaz de Paso de Mensajes) con una plataforma de memoria distribuida.

4. Resultados Experimentales

Los resultados obtenidos en esta implementación paralela serán evaluados en términos de:

- Tiempo de ejecución: Es el tiempo que lleva solucionar el problema.
- Speed-up: Se define como la proporción de tiempo que toma solucionar el problema sobre un procesador y el tiempo requerido para solucionar el mismo problema sobre un computador paralelo con p procesadores idénticos.
- Eficiencia: Es la medida de fracción de tiempo durante el cual un procesador es empleado útilmente; es definido como la proporción entre el Speed-up y el número de procesadores.

En las figuras 2, 3, 4 y 5 se presentan los gráficos de: Tiempos de Ejecución, Speed-up, Eficiencia y Evaluaciones de Distancia (E.D.) respectivamente. Los resultados corresponden a la estructura *spaghettis* construida para 4, 8, 12, 16 y 20 pivotes y búsquedas con rango 1, 2, 3 y 4. Los resultados se presentan para el caso secuencial (1 procesador) y la implementación paralela (2, 4 y 8 procesadores).

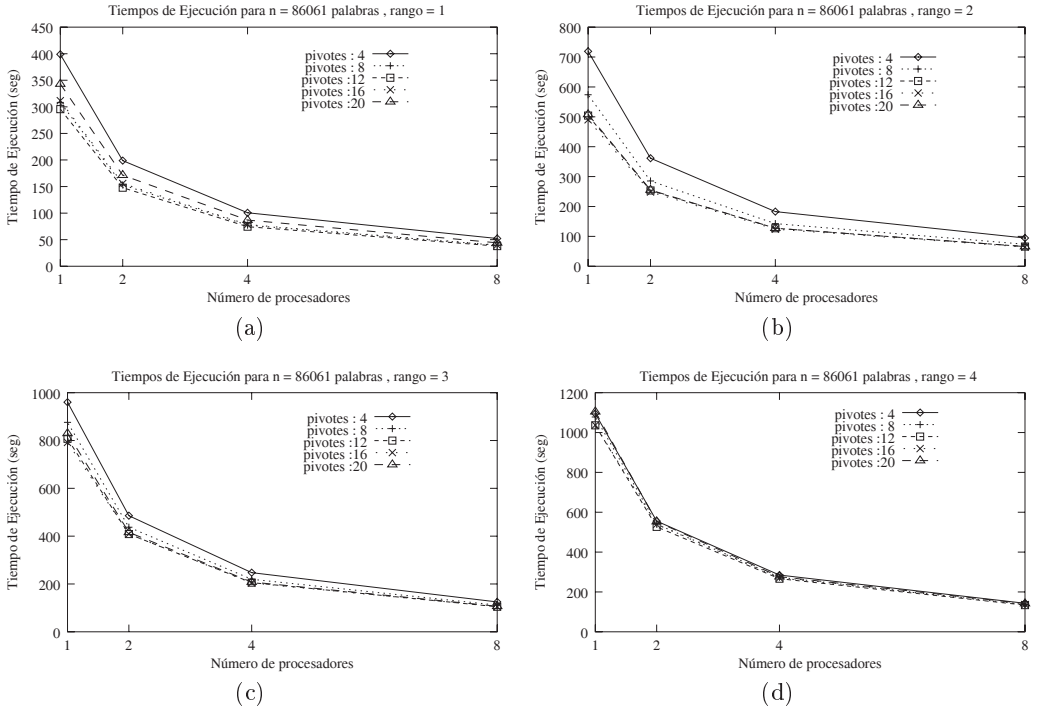


Figura 2. Gráficos de Tiempos de Ejecución

5. Conclusiones

5.1. Aspectos Relevantes y Aportes

La mayor parte de las estructuras que se han desarrollado para búsquedas por similaridad estan diseñadas sólo como prototipos para memoria principal. En este sentido, se considera que el aporte más relevante de éste artículo es la presentación de una versión paralelizada de la búsqueda sobre la estructura *Spaghettis* y optimizada para memoria secundaria, lo que permite contar con una estructura que puede ser preparada para aplicaciones reales.

De acuerdo con los resultados experimentales los tiempo de ejecución y las evaluaciones de distancia se han reducido notablemente para la estructura construída con cada número de pivotes y los rangos de búsquedas utilizados para estos. En la utilización de 4 pivotes versus 8 pivotes y la realización de búsquedas para rangos 1 y 2 (ver Figuras 2(a) y 2(b)) es donde más se acentúa esta disminución de tiempo.

En general los costos de tiempo de ejecución se han reducido significativamente debido al balance de consultas que existe, donde cada procesador tiene acceso al mismo número de estas. Por otra parte, la reducción de *evaluaciones de distancia*, el proceso mas costoso dentro de la búsqueda, influye en gran medida en la reducción de estos tiempos. En la Tabla 1 se

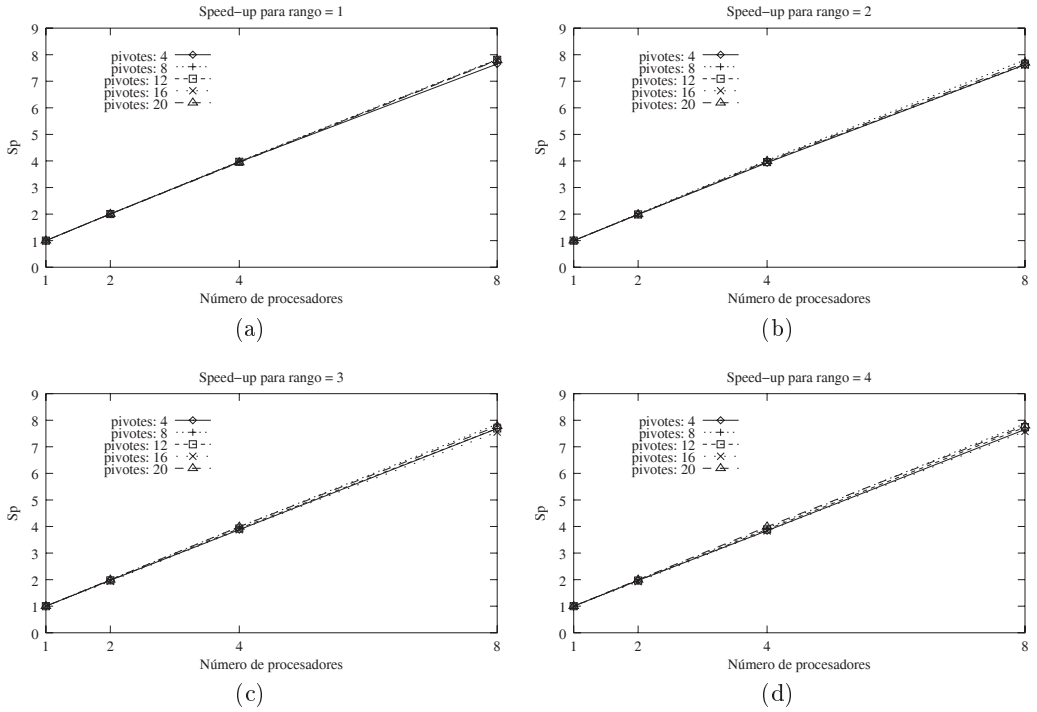


Figura 3. Gráficos de Speed-up

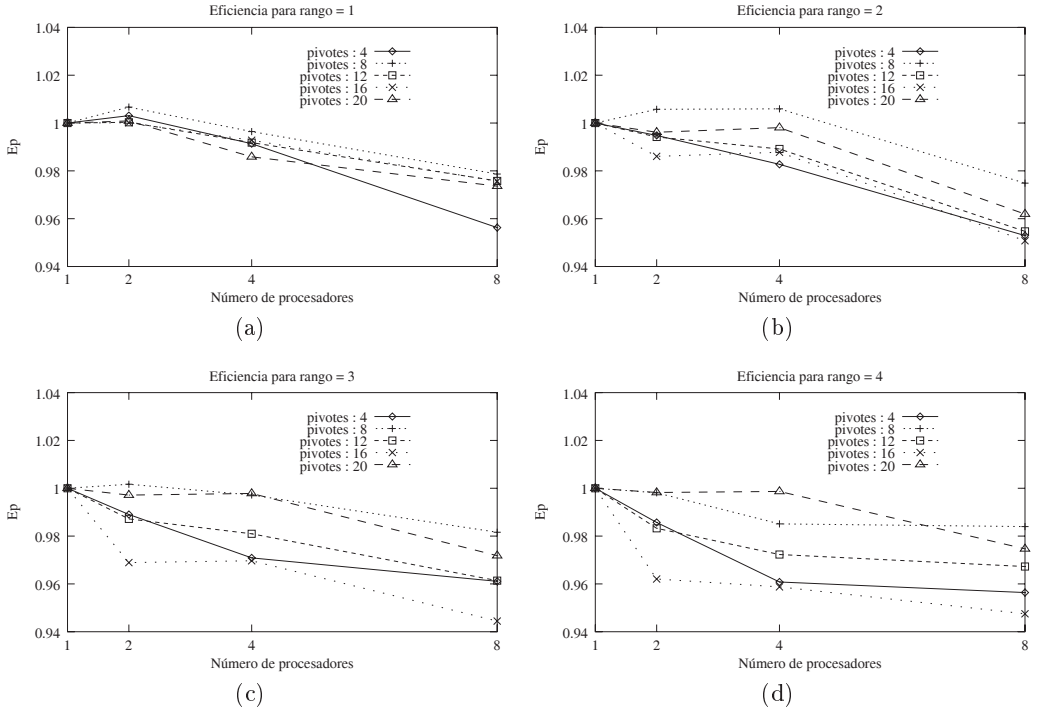


Figura 4. Gráficos de Eficiencia

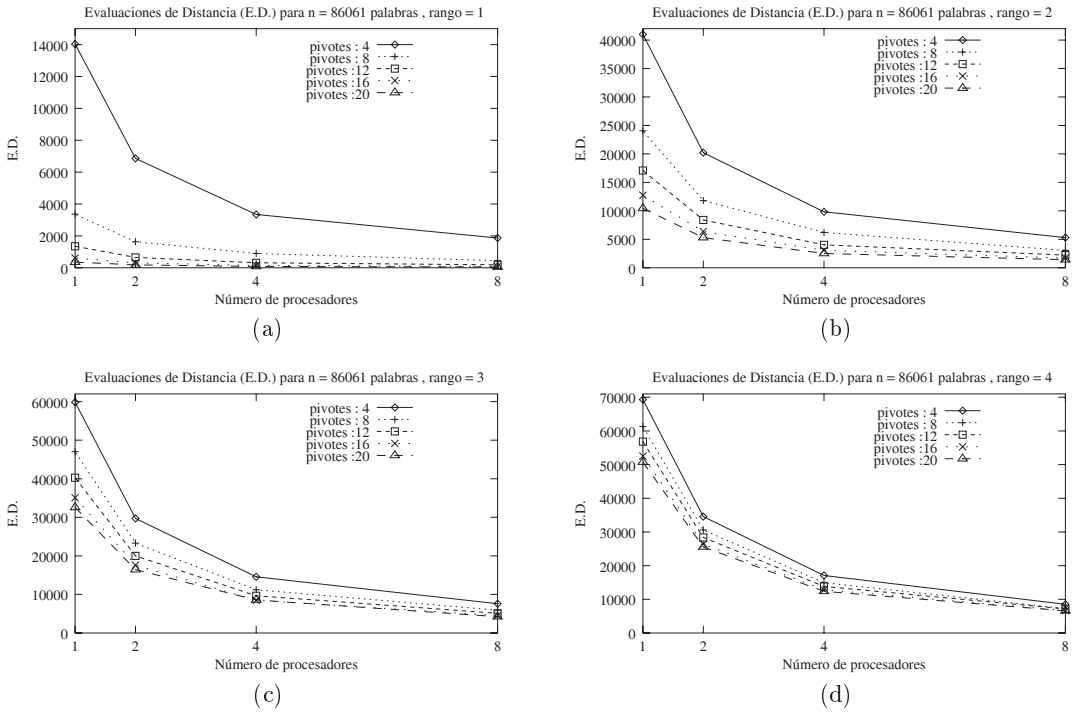


Figura 5. Gráficos de Evaluaciones de Distancia (E.D.)

presentan los porcentajes de disminución de evaluaciones de distancia de acuerdo al número de procesadores utilizado.

num. procs	4 Pivotes	8 Pivotes	12 Pivotes	16 Pivotes	20 Pivotes
2	50.6 %	50.8 %	50.8 %	49.8 %	49.3 %
4	75.7 %	74.8 %	76.2 %	75.9 %	75.4 %
8	87.2 %	87.4 %	87 %	86.7 %	86.6 %

Cuadro 1. Promedios de disminución de Evaluaciones de Distancia.

Conforme aumenta el rango de búsqueda los tiempos de ejecución para todos los pivotes es muy similar llegando incluso a ser casi idénticos para las búsquedas de rango 4 (ver Figura 2). Esto se debe a que para consultas de mayor rango son más los objetos que se recuperaran, lo que implica recorrer en mayor medida la estructura.

Por último, teniendo en cuenta la Figura 4, se puede apreciar que la estructura construída mediante 8 pivotes es quien mejor se comporta con respecto al algoritmo secuencial. Esto para cada número de procesadores y rangos de búsquedas utilizados.

6. Trabajos Futuros

- Emplear heurísticas para la selección de pivotes. Entre estas, las propuestas de elegir objetos más alejados entre sí y la selección de objetos espacialmente esparcidos.
- Probar el comportamiento de la estructura con otros tipos de espacios métricos.
- Implementar métodos de eliminación y en general modificar la estructura para que sea dinámica.

- Implementar distintas alternativas de distribución de la base de datos sobre la estructura paralela.

7. Agradecimientos

Lo autores desean dar expresos agradecimientos a los grupos *RETICS* (Sistemas Concurrentes y Tiempo Real) y *SIMD* (Sistemas Inteligentes y Minería de Datos) del *Instituto de Investigación Informática de Albacete* (*I3A*) de la Universidad de Castilla La Mancha, España por las facilidades otorgadas en el uso de sus instalaciones.

Referencias

1. R. Baeza-Yates, W. Cunto, U. Manber, and S. Wu. Proximity matching using fixedqueries trees. In *5th Combinatorial Pattern Matching (CPM'94)*, LNCS 807, pages 198–212, 1994.
2. Sergei Brin. Near neighbor search in large metric spaces. In *the 21st VLDB Conference*, pages 574–584. Morgan Kaufmann Publishers, 1995.
3. W. Burkhard and R. Keller. Some approaches to best-match file searching. *Communication of ACM*, 16(4):230–236, 1973.
4. E. Chavéz, J. Marroquín, and R. Baeza-Yates. Spaghettis: An array based algorithm for similarity queries in metric spaces. In *6th International Symposium on String Processing and Information Retrieval (SPIRE'99)*, pages 38–46. IEEE CS Press, 1999.
5. P. Ciaccia, M. Patella, and P. Zezula. M-tree : An efficient access method for similarity search in metric spaces. In *the 23st International Conference on VLDB*, pages 426–435, 1997.
6. W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with Message-Passing Interface*. MIT Press, 1994.
7. L. Micó, J. Oncina, and E. Vidal. A new version of the nearest-neighbor approximating and eliminating search (aesa) with linear preprocessing-time and memory requirements. *Pattern Recognition Letters*, 15:9–17, 1994.
8. Gonzalo Navarro. Searching in metric spaces by spatial approximation. *The Very Large Databases Journal (VLDBJ)*, 11(1):28–46, 2002.
9. Nora Reyes. Índices dinámicos para espacios métricos de alta dimensionalidad. Master's thesis, Universidad Nacional de San Luis, Argentina, 2002.
10. J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. In *Information Processing Letters*, pages 40:175–179, 1991.
11. Roberto Uribe-Paredes. Manipulación de estructuras métricas en memoria secundaria. Master's thesis, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Santiago, Chile, Abril 2005.
12. Roberto Uribe-Paredes and Christian Cárdenas. Spaghettis en memoria secundaria. In *XII Congreso Argentino de Ciencias de la Computación (Cacic2006)*, San Luis, Argentina, Oct. 2006.
13. P. Yianilos. Data structures and algoritms for nearest neighbor search in general metric spaces. In *4th ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, pages 311–321, 1993.

Restricciones en la replanificación del mantenimiento de locaciones petroleras

A. Villagra, C. Montenegro, E. de San Pedro, M. Lasso, D. Pandolfi*

LabTEm - Unidad Académica Caleta Olivia
Universidad Nacional de la Patagonia Austral
Caleta Olivia (9011) – Santa Cruz - Argentina
{avillagra, cmontenegro, edesanpedro, mlasso, dpandolfi}@uaco.unpa.edu.ar

Abstract

This work deals with an application (PAE) that uses an evolutionary algorithm, generator of multiple solutions for the rescheduling of preventive maintenance of oil fields. The exploitation and the transport of oil are very important activities for the economic development of the industrial modern society. However, these activities are generating risks that are translated in contaminations. It is important that oil companies carry out a correct maintenance of their oil fields. PAE provides a maintenance visit schedule at the right moment and this benefit should be observed from two aspects. First, a schedule is better than other if for the same number of oil wells to be visited, the round cost and the time scheduled is small. Second, if for the same period of time it is possible to carry out the maintenance to more oil wells, this benefit reduces the probability of falling when the number of visited oil wells is increased. PAE is in the fourth phase of its development, in which we have incorporated restrictions to the moment of rescheduling. The obtained results have been satisfactory since they minimize the total time of a scheduling and they maximize the quantity of visited oil wells, satisfying the constraints.

Keywords: **Oil Fields, constraints, multirecombination, evolutionary algorithm.**

Resumen

En este trabajo se presenta una herramienta (PAE) que utiliza un algoritmo evolutivo, generador de múltiples soluciones, para la replanificación dinámica del mantenimiento preventivo de locaciones petroleras. La explotación y el transporte de petróleo son actividades muy importantes para el desarrollo económico de la sociedad industrial moderna. Sin embargo, estas actividades son generadoras de riesgos que se traducen en contaminaciones. Es importante que las empresas petroleras realicen un correcto mantenimiento de sus locaciones. PAE brinda en forma oportuna la planificación del recorrido y su beneficio debe observarse desde dos aspectos. Primero, una planificación es mejor que otra, si para un mismo número de locaciones a visitar el costo de recorrido e intervención planificado es menor. Segundo, si con un mismo tiempo de intervención es posible realizar el mantenimiento a más locaciones, este beneficio reduce la probabilidad de caída de la producción al incrementar la cantidad de locaciones recorridas. PAE está en la cuarta fase de su desarrollo, donde se han incorporado restricciones al momento de la replanificación. Los resultados obtenidos han sido satisfactorios ya que minimizan el tiempo total de una planificación y maximizan la cantidad de locaciones visitadas, satisfaciendo las restricciones.

Palabras claves: **Locaciones petroleras, restricciones, multirecombinación, algoritmo evolutivo.**

* El Grupo de Investigación cuenta con el apoyo de la Universidad Nacional de la Patagonia Austral

1 INTRODUCCIÓN

El petróleo, recurso natural de gran importancia para el desarrollo de la humanidad, y el empleo de tecnología, como instrumento de apropiación y de transformación de los recursos naturales, han generado en el mundo impactos ambientales negativos, significativos por las graves consecuencias provocadas en el medio. Este recurso se ha convertido en una preocupación ambiental seria, dado que su extracción y uso como fuente de energía por los seres humanos, ha conducido a su distribución amplia en la biosfera. En muchas de las regiones donde se ha explotado el petróleo, se han generado contingencias de diferentes magnitudes como consecuencia de las operaciones desarrolladas en: tanques de almacenamiento, oleoductos, equipos de perforación, buques petroleros, etc. Además de los impactos generados en el medio como consecuencia de la exploración y explotación del petróleo, existen también daños ocasionados a las personas que lo emplean o se encuentran en contacto con él dentro de su ámbito laboral.

En la última década, las Empresas dedicadas a la explotación, producción y transporte del recurso, han visto la necesidad de implementar medidas de prevención a fin de evitar y / o minimizar los daños ocasionados al medio ambiente, personas y bienes materiales.

La mejor manera de atacar el problema de contaminación por petróleo es prevenir el incidente. Normalmente estos se producen por fallas de equipos o del material y fallas humanas. Los primeros pueden ser subsanados mediante inspecciones periódicas y un mantenimiento adecuado; y los segundos, mediante la instrucción y el entrenamiento del personal en forma permanente. Por esta razón es importante, para las empresas petroleras y para el entorno que las rodea, un correcto mantenimiento de sus locaciones.

Los Algoritmos Evolutivos (AEs) son metaheurísticas que emplean modelos computacionales que simulan el proceso evolutivo. Existen una gran variedad de AEs, los principales incluyen: Algoritmos Genéticos [9] y [11], Programación Evolutiva [7] y [8], Estrategias Evolutivas [16] y [17], y Programación Genética [12]. Todos estos algoritmos comparten un concepto base común que es simular a la evolución de los individuos que forman la población usando un conjunto de operadores predefinidos. Comúnmente se usan dos tipos de operadores: de selección y de búsqueda. Los operadores de búsqueda más usados son la mutación y la recombinación. Tendencias actuales en AEs hacen uso de enfoques con multirecombinación [1], [2] y [3], y múltiples padres [4], [5] y [6]. Para la resolución de diversos tipos de problemas de planificación tales como *scheduling* o *routing* estos enfoques han resultado ser estrategias exitosas. Particularmente en problemas de *scheduling*, introduciendo al enfoque de multirecombinación una nueva variante conocida como MCMP-SRI (*Stud and Random Immigrates*) [14].

La herramienta presentada aquí utiliza un algoritmo evolutivo que es el generador de múltiples soluciones. Se encuentra en la cuarta fase de desarrollo del prototipo en la que se han incorporado restricciones al momento de la replanificación. Avances de fases anteriores han sido presentados en otros trabajos [18], [19] y [20].

El trabajo está organizado de la siguiente manera, en la sección 2 se presenta la descripción del problema y su definición formal. En la sección 3 se describe el algoritmo utilizado para la obtención de una replanificación, en la sección 4 se muestran los detalles de implementación y los resultados obtenidos. Finalmente, en la sección 5 se describen las conclusiones.

2 DOMINIO Y DESCRIPCIÓN DEL PROBLEMA

Las empresas petroleras realizan visitas de mantenimiento preventivo a cada una de sus locaciones petroleras (pozos productores, inyectores, baterías y colectores). Un yacimiento está formado por bloques y a su vez éste por baterías. Cada batería está formada por pozos de producción que son en promedio entre 15 y 20. Cada pozo tiene diferente nivel de producción que es conocido a priori y varía en el tiempo. La producción del pozo define la categoría y la cantidad de veces que debe visitarse al mes. Los pozos no pueden ser visitados más de una vez al día y dependiendo del tipo de pozo, existen ciertas tareas que se deben realizar. Cada tarea tiene asignado un determinado equipamiento necesario, una frecuencia de realización y un tiempo aproximado de su duración. En la Tabla 1 se muestran ejemplos de algunas tareas realizadas en una locación, en este caso, tareas en pozos productores y baterías. Actualmente, el recorrido que realizan los encargados de las locaciones, se planifica en base a la experiencia de los mismos. La jornada laboral comienza a la mañana y se visitan las locaciones en dos turnos de tres horas. Luego de finalizado cada turno el responsable debe regresar a la base de operaciones, realizar determinadas actividades administrativas y luego comenzar con el siguiente turno. El tiempo demandado en cada locación dependerá del tipo de la misma. Existen contingencias aleatorias que hacen que el plan de mantenimiento de un turno no se cumpla según lo planificado, produciendo la necesidad de replanificar las visitas. Además puede ocurrir que en el momento de replanificación, por determinados motivos, se debe incorporar la visita obligatoria a ciertas locaciones en el próximo turno, provocando esto un conjunto de restricciones en la replanificación. Cuando ocurre esto, cada responsable redefine el nuevo itinerario utilizando su experiencia. En la Figura 1 se muestra una distribución de locaciones petroleras del yacimiento explotado en la zona norte de la provincia de Santa Cruz, Argentina.

Tabla 1. Tareas en un pozo productor y baterías

descripción de la tarea	Cant. Equipos	Frecuencia/días	Tiempo/Min.
Verificación régimen de bombeo	78	7	2
Extracción muestra boca de pozo (por pozo)	105	15	5
Medición de gas de entrecaños	5	30	10
Puesta de ensayos en auxiliares y Estación	12	1	10
Medición y cierre de Tks en estaciones	6	1	10
Verificación de recirculado	3	1	5
Verificación de bombas de baterías y Stock	4	7	5
Verifiación de inyección de química Bat y pozos	15	1	5
Estado de puente de producción	105	1	5
Programa semanal de Dinamometro y nivel	78	7	20

PAE tiene por objetivos planificar las visitas a un conjunto de locaciones que:

- Minimice el tiempo total de visitas, es decir, encontrar la planificación que en menor tiempo recorra las locaciones incluyendo el tiempo de intervención en cada una de las mismas.
- Maximice la cantidad de visitas en un período de tiempo. Teniendo en cuenta un período de mantenimiento, por ejemplo de un mes, que en ese período visite la mayor cantidad de veces las locaciones.
- Replanifique las visitas a partir de desviaciones en la planificación original. Frente a la ocurrencia de eventos que condicionan operativamente la ejecución de un plan de mantenimiento, proveer de planificaciones alternativas sin disminuir significativamente la calidad de las mismas. Cumplimente un conjunto de restricciones que se le incorporan al momento de la replanificación.

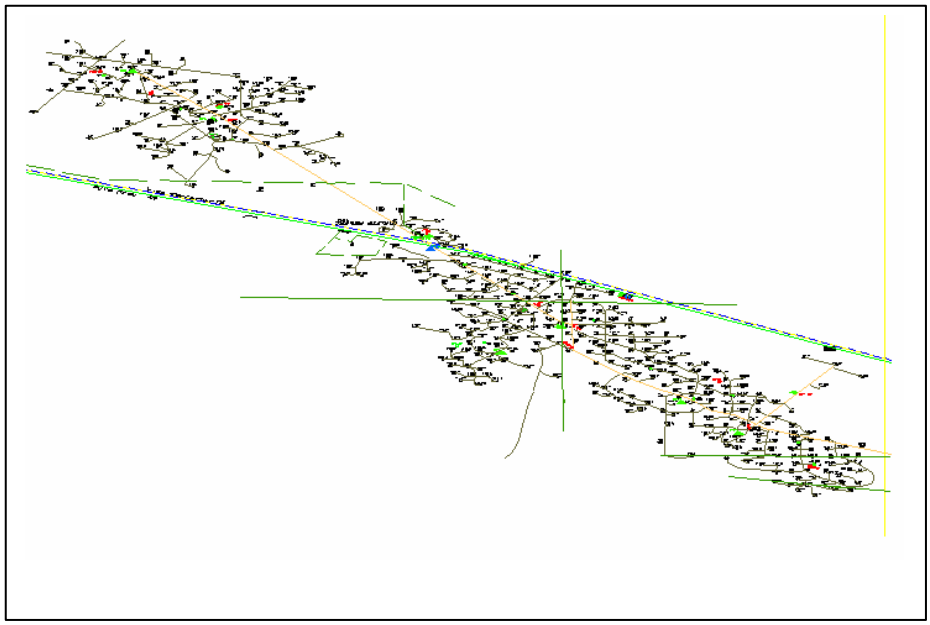


Figura 1. Plano de caminos y distribuciones

2.1 Formulación del Problema

El problema se puede definir como [15]: $1|s_{jk}|C_{\max}$

Denota un problema de *scheduling* de máquina única con n tareas sujetas a tiempos de preparación dependientes de la secuencia. Donde las tareas a planificar son el servicio de mantenimiento (o intervención) en cada una de las locaciones petroleras. Además, existe un tiempo de traslado entre cada una de las locaciones al que se denomina s_{jk}, que representa el costo en tiempo de ir de la locación j a la locación k.
La función objetivo es minimizar el *makespan* (C_{max}) sujeto a los tiempos de preparación dependientes de la secuencia. Este problema es equivalente al denominado *Traveling Salesman Problem* (TSP).

3 ALGORITMO EVOLUTIVO PARA PAE

Para resolver el problema de planificación de recorrido de las locaciones petroleras, se utilizó un algoritmo evolutivo. Para codificar adecuadamente las visitas a las locaciones petroleras que representan una posible solución, se utilizó una permutación. Donde cada permutación $p = (p_1, p_2, \ldots, p_n)$ es un cromosoma en el cual p_i representa la locación i que debe ser visitada y n representa la cantidad de locaciones a visitar. El cromosoma establece el orden de la secuencia a seguir para visitar cada locación. Además, se tiene en cuenta que existen locaciones que deben ser visitadas obligatoriamente en el próximo turno, interrumpiendo la secuencia de la planificación original. Esto implica incorporar restricciones al problema. Para ello, se definió un tipo de restricción llamada restricción dura, donde toda solución obtenida que no cumpla con este tipo de restricción es considerada no factible y por lo tanto debe ser reparada o eliminada. Para este caso, la restricción establece que en el momento de la replanificación y en el próximo turno se visiten obligatoriamente ciertas locaciones.

En la Figura 2 se puede observar la estructura del algoritmo evolutivo utilizado para este problema de planificación. El algoritmo crea una población inicial *Stud(0)* de sementales generados aleatoriamente que representan μ soluciones al problema. Luego estas soluciones se evalúan donde se analiza si la planificación obtenida cumple con la restricción. En caso de no cumplir la restricción, la solución se repara de la siguiente manera: se desplazan las locaciones que no están correctamente planificadas, y se ubican en el turno correspondiente las locaciones que deben visitarse obligatoriamente en ese turno (ver Figura 3). Finalmente se evalúa la solución. Seguidamente, de la población de individuos denominados sementales, se selecciona un individuo a través de selección proporcional. Se genera un *pool* de apareamiento con *n2* padres generados aleatoriamente (inmigrantes). El semental se aparea con cada padre inmigrante del *pool* de apareamiento y las parejas se someten a operaciones de recombinación, y se generan *2*n2* descendientes. El mejor de los *2*n2* descendientes, se almacena en un *pool* de hijos temporal. Esta operación de recombinación se repite *n1* veces, para diferentes puntos de corte cada vez, hasta que el *pool* de hijos se complete. Finalmente, el mejor descendiente creado de *n2* padres y *n1* operaciones de recombinación, se inserta en la nueva población.

```
EA-MCMP-SRI
t=0; {generación actual}
iniciliza (Stud(t));
evalua (Stud(t));
while (not max_evaluations) do
   pool_aparemiento = Genera_Inmigrantes_Aleatorios
                          ∪ Seleciona (Stud(t));
   while (not n2) do
        while (not  n1) do
            evoluciona (pool_apareamiento);
                 {recombinación y mutación}
        end while
   end while
   evalua (pool_apareamiento);
   Stud(t+1)= Selec. nueva población del pool_apareamiento;
    t = t+1;
end while
```

Figura 2. Algoritmo Evolutivo para PAE

Este proceso es llamado MCMP-SRI y es una variante de multirecombinación [14]. Fue aplicado en diferentes problemas de planificación de máquina única para casos estáticos y casos dinámicos y los resultados obtenidos fueron satisfactorios.

El método de recombinación utilizado fue PMX (*Partial Mapped Crossover*). Este operador fue propuesto por [10] y puede verse como una extensión del cruzamiento de dos puntos para representaciones basadas en permutaciones.

La operación de mutación usada es llamada *Swapping Mutation* (SM), la cual selecciona dos posiciones aleatorias e intercambia los alelos. La selección de individuos fue a través del operador de selección proporcional.

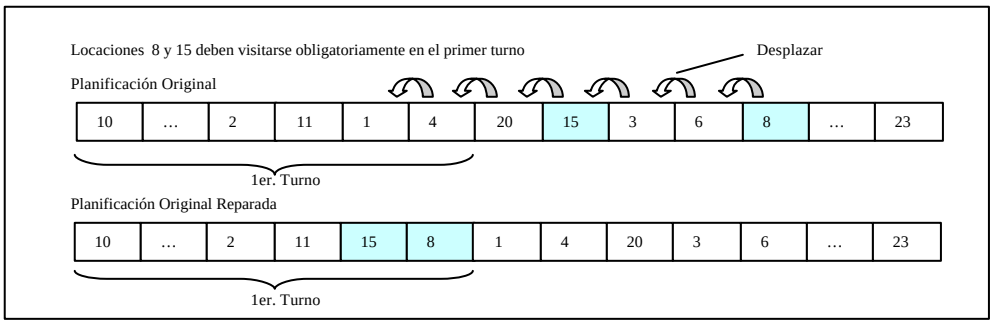


Figura 3. Reparación de la solución

4 EXPERIMENTOS Y RESULTADOS

4.1 Preparación de Datos

Para resolver el problema fue necesario preparar los datos de entrada ya que originalmente las distancias entre las locaciones petroleras no estaban procesadas. Se realizó el cálculo de las distancias entre las locaciones petroleras basados en el plano de caminos y distribución del yacimiento. Matemáticamente, es sabido que la distancia entre dos puntos que se encuentran en cualquier lugar del sistema de coordenadas, está determinada por la relación denominada distancia euclídea. No obstante, en este problema solo se puede calcular la distancia entre dos puntos, teniendo en cuenta el camino que existe para llegar a ellos. Por esta razón se utilizó el plano de las ubicaciones de las locaciones y se escalaron las distancias entre las mismas.

4.2 Diseño Experimental y Resultados

Para realizar los experimentos se establecieron las siguientes suposiciones y restricciones al problema. Para la evaluación de la aplicación se trabajó con 110 locaciones petroleras correspondientes a un bloque de la zona de explotación. La velocidad de recorrido se estableció en 12 segundos cada 100 metros y se fijó el mismo tiempo de intervención para cada locación en el proceso de mantenimiento preventivo.

Para el algoritmo evolutivo propuesto se utilizó un tamaño de población de 15 individuos. La población inicial se generó aleatoriamente. El número máximo de evaluaciones es 2^{22}. Se estableció la probabilidad de mutación en 0,05 y la probabilidad de recombinación en 0,65. El número *n1* (número de operaciones de recombinación) y *n2* (número de padres), se estableció en 16 y 18 respectivamente. Los parámetros (tamaño de la población, criterio de parada, probabilidades, etc.) se seleccionaron en base a la experimentación de los valores previamente usados con éxito [13]. La Tabla 2 resume los parámetros utilizados en todas las corridas.

Para analizar su eficiencia, se realizaron cuatro tipos de experimentos y en todos los casos la velocidad de recorrido se estableció en 12 segundos cada 100 metros y se fijó el mismo tiempo de intervención para cada locación en el proceso de mantenimiento preventivo. El primer experimento se realizó con el objetivo de analizar la eficiencia del algoritmo para la planificación del total de locaciones correspondientes al bloque norte de la zona de explotación (110 locaciones) y los restantes experimentos, con el objetivo de analizar la eficiencia del algoritmo en cuanto a la replanificación e incorporación de visitas obligatorias a ciertas locaciones en el próximo turno.

Tabla 2. Parámetros del algoritmo evolutivo

Tamaño población	15
Tamaño cromosoma (cantidad de locaciones)	110
Criterio parada (generación)	500
Recombinación	PMX
Mutación	SW
Probabilidad Recombinación	0,65
Probabilidad Mutación	0,05
Nº de recombinación (n1)	16
Nº de padres (n2)	18

Tabla 3. PAE sin restricciones

Nº	**Tr.**	**Km**	**Min**	**Tpo.Total**
1	5	112,50	774	12:54:02
2	5	114,83	778	12:58:37
3	5	115,84	780	13:00:42
4	5	125,16	799	13:19:22
5	5	119,87	788	13:08:44
6	5	110,87	770	12:50:41
7	5	133,28	815	13:35:33
8	5	104,80	758	12:38:30
9	5	120,99	791	13:10:59
10	5	**104,12**	**757**	**12:37:19**
11	5	111,96	772	12:52:57
12	5	124,86	798	13:18:37
13	5	118,38	785	13:05:45
14	5	107,96	764	12:44:55
15	5	119,78	788	13:08:34
16	5	126,68	802	13:22:21
17	5	122,24	792	13:12:59
18	5	119,16	787	13:07:18
19	5	**133,66**	**816**	**13:36:20**
20	5	119,79	788	13:08:37
Mínimo		**104,12**	**757**	
Máximo		**133,66**	**816**	

En todos los experimentos se realizaron 20 corridas independientes. En el primer experimento se usaron los valores de los parámetros resumidos en la Tabla 2, en los experimentos restantes se modifico únicamente el tamaño de las locaciones a visitar, y los valores restantes se mantuvieron. A continuación se describe cada uno se los experimentos con replanificación:

- Se tomaron 110 locaciones a visitar, con 2, 3, 4 y 5 visitas obligatorias en un turno.
- Se tomaron 80 locaciones a visitar, con 2, 3, 4 y 5 visitas obligatorias en un turno
- Se tomaron 55 locaciones a visitar, con 2, 3, 4 y 5 visitas obligatorias en un turno.

En la Tabla 3 se muestran los resultados obtenidos para una planificación sin restricciones por el algoritmo propuesto. La primera columna representa el número de corrida, la segunda columna indica la cantidad de turnos planificados, la tercera columna muestra los kilómetros recorridos en esa planificación, la cuarta columna muestra los minutos totales correspondientes a la planificación y finalmente la última columna detalla el tiempo total en horas. Las dos últimas filas muestran los valores mínimos y máximos obtenidos. Se puede observar que en cada corrida la cantidad de turnos planificados es de 5. El tiempo total planificado en cada una de las corridas para realizar las visitas a las 110 locaciones varía entre 12:37 y 13:36 horas. El mínimo tiempo para una planificación se encontró en la corrida 10 donde se visitan las 110 locaciones en 757 minutos (12 horas, 37 minutos y 19 segundos) y se recorren 104,12 km. El máximo tiempo planificado se encontró en la corrida 19 donde se visitan las 100 locaciones en 816 minutos (13 horas, 36 minutos y 20 segundos) y se recorren 133,66 km.

En la Tabla 4 se muestra una comparación de una planificación específica para el recorrido de 110 locaciones, realizada por la empresa petrolera y la planificación obtenida por PAE para la misma cantidad de locaciones. Se puede observar que PAE reduce en un 30 % el tiempo total de planificación y visita en todos los turnos más cantidad de locaciones que las visitadas por la empresa petrolera. En general, mientras una planificación original demanda un tiempo total de 18 horas 24 minutos, la mejor planificación provista por PAE demanda 12 horas 37 minutos, logrando un ahorro de aproximadamente 6 horas.

Tabla 4. Comparación Empresa Petrolera y PAE

Planificación Empresa Petrolera				Planificación con PAE			
Día	**Turno**	**Pozos**	**Tiempo**	**Día**	**Turno**	**Pozos**	**Tiempo**
1	1	18	181,50	1	1	24	178,67
	2	16	176,23		2	27	182,02
2	3	18	184,20	2	3	26	177,95
	4	20	182,20		4	27	178,97
3	5	19	181,90	3	5	6	39,72
	6	19	198,60			**110**	**757,16**
		110	**1104,63**				**12:37:19**
			18:24:38				

Tabla 5. Replanificación con restricciones

Locaciones	Visitas Obligatorias	Km	Turnos	Tpo.Total
110	2	124,333	5	13:17:41
	3	120,888	5	13:10:47
	4	129,147	5	13:27:19
	5	126,681	5	13:22:23
80	2	89,379	4	9:37:16
	3	83,511	4	9:24:28
	4	87,821	4	9:34:39
	5	82,802	4	9:23:07
55	2	58,044	3	6:29:53
	3	54,360	3	6:22:10
	4	51,465	3	6:15:19
	5	53,560	3	6:10:07

En la Tabla 5 se muestran los resultados obtenidos por PAE en cada uno de los experimentos descriptos anteriormente donde se realiza replanificación y se incorporan restricciones. La primera columna corresponde a la cantidad de locaciones a visitar en el momento de la replanificación. La segunda columna corresponde a la cantidad de locaciones que obligatoriamente se deben visitar en el próximo turno. La tercera columna corresponde a los kilómetros recorridos en esa replanificación. La cuarta columna corresponde a la cantidad de turnos a realizar y finalmente la última columna corresponde al tiempo total de esa replanificación, en horas. Se puede observar que la incorporación de restricciones al momento de la replanificación, ya sean 2, 3, 4 ó 5 locaciones, no degrada la eficiencia de los resultados obtenidos pues en todos los casos el algoritmo continúa mejorando las planificaciones realizadas por la empresa petrolera.

5 CONCLUSIONES

PAE es una aplicación que tiene como objetivo ser una herramienta eficaz que facilite la planificación dinámica del mantenimiento de locaciones petroleras. Del análisis y las comparaciones realizadas con los planes de mantenimiento ejecutados, PAE ofrece las siguientes ventajas comparativas:

- En cuanto a la calidad de las soluciones, PAE presenta planificaciones que mejoran el plan de mantenimiento producido por expertos, reduciendo el tiempo total de una planificación tipo, con la correspondiente reducción de costos. Sin embargo, éste beneficio puede también analizarse desde otra perspectiva, ya que reduciendo el tiempo total de intervención, se puede por lo tanto realizar más cantidades de visitas en las locaciones en un determinado

período. Con ello se logra disminuir la probabilidad de caída de la producción y por lo tanto maximizar la producción total.

- Los AEs son algoritmos estocásticos (no determinísticos) que producen múltiples soluciones en diferentes corridas independientes. A menudo una solución mejor (plan de mantenimiento) no puede ejecutarse por determinadas condiciones operativas, por lo tanto es necesario seleccionar otra que si bien puede no ser tan buena como la anterior es factible de ejecutarse.
- Otro aspecto que suele ser muy importante, es la flexibilidad de producción de planes de mantenimiento, ya que muy a menudo se producen cambios, incorporando o eliminando locaciones en la producción del yacimiento. Para ello PAE facilita un ambiente flexible que permite incorporar cambios en la planificación sin que ello represente la intervención de expertos.
- Por último, frente a la incorporación de restricciones en las visitas de mantenimiento, PAE no presenta degradación en la solución, generando planificaciones que satisfacen las restricciones y mantienen la calidad de los resultados.

Trabajos futuros incluirán el aprendizaje de contingencias en el mantenimiento.

AGRADECIMIENTOS

Se reconoce a la Universidad Nacional de la Patagonia Austral por su apoyo al grupo de investigación y la cooperación y las críticas constructivas proporcionadas por el mismo.

REFERENCIAS

[1] Eiben A.E. y Bäck Th. "An Empirical Investigation of Multi-Parent Recombination Operators in Evolution Strategies", *Evolutionary Computation*, 5(3):347-365, (1997).

[2] Eiben A.E., Raué P.E. y Ruttkay Z., "Genetic algorithms with multi-parent recombination". *Proc. 3rd Conference on Parallel Problem Solving from Nature*, Springer-Verlag, number 866 in LNCS, pp. 78-87, (1994).

[3] Eiben A.E., Van Kemenade C.H.M., y Kok J.N. "Orgy in the Computer: Multi-Parent Reproduction in Genetic Algorithms". *Proc. 3rd European Conference on Artificial Life*, Springer-Verlag, number 929 in LNAI, pages 934-945, (1995).

[4] Esquivel S., Leiva A., Gallard R. "Multiple Crossover per Couple in Genetic Algorithms". *Proc. Fourth IEEE Conference on Evolutionary Computation (ICEC'97)*, pp 103-106, Indianapolis, USA April (1997).

[5] Esquivel S., Leiva A., Gallard R. "Couple Fitness Based Selection with Multiple Crossover per Couple in Genetic Algorithms". *Proc. International Symposium on Engineering of Intelligent Systems (EIS´98)*, pp 235-241, La Laguna, Tenerife, Spain, February (1998).

[6] Esquivel S., Leiva H., Gallard R. "Multiple Crossovers between Multiple Parents to Improve Search in Evolutionary Algorithms". *Proc. Congreso on Evolutionary Computation (IEEE)*, pp 1589-1594, Washington DC, (1999).

[7] Fogel D. Evolutionary Computation: Towards a NewPhilosophy of Machine Intelligence. *IEEE Press*, Piscataway, NJ, 1996.

[8] Fogel L.J., Owens A.J., y Walsh M.J. *Artificial intelligence through simulated evolution*. Wiley, 1966.

[9] Goldberg. D. *"Genetic Algorithms in Search, Optimization and Machine Learning"*. Addison Wesley, Reading, MA, (1989).

[10] Goldberg, D. y R. Lingle, "Alleles, loci and the traveling salesman problem". *Proc. of the First International Conference on Genetic Algorithms*, Lawrence Eribaum Associates, pp. 154-159, Hillsdale, NJ, (1987).

[11] Holland. J.H. *"Adaptation in natural and artificial system"*. University of Michigan Press, 1975. New York, (1995).

[12] Koza J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[13] Lasso M., Pandolfi D, De San Pedro M.E., Villagra A., Gallard R.; "Solving Dynamic Tardiness Problems in Single Machine Environments"; *Congress on Evolutionary Computation - CEC '04*; Portland, U.S.A, Vol 1, 1143-1149, (2004).

[14] Pandolfi D., De San Pedro M., Villagra A., Vilanova G., Gallard R. "Studs Mating Immigrants in Evolutionary Algorithm to Solve the Earliness-Tardiness Scheduling Problem". *Cybernetics and Systems of Taylor and Francis Journal*, Volume 33 (U.K.), pp 391-400, (2002).

[15] Pinedo M., Scheduling: *Theory, Algorithms and System*, First edition Prentice Hall, (1995).

[16] Rechenberg I., Evolution strategy. In J.M. Zurada, R.J. Marks II, and C. Robinson, editors, *Computational Intelligence: Imitating Life*. IEEE Press, Piscataway, NJ, 1994.

[17] Schwefel H.P. *Evolution and Optimum Seeking*. Wiley, New York, 1995.

[18] Villagra A., de San Pedro M., Lasso M., Montenegro C., Pandolfi D. ,"Evolutionary Algorithm for the Oil Fields Preventive Maintenance Scheduling", *The 11th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, Florida, USA (2007).

[19] Villagra A., Montenegro C., Rasjido J, de San Pedro M., Lasso M., Pandolfi D.; "PAE: una herramienta para la planificación del mantenimiento en locaciones petroleras"; *XII Congreso Argentino de Ciencias de la Computación – CACIC 2006*; Universidad Nacional de San Luis, San Luis; (2006).

[20] Villagra A., Montenegro C., de San Pedro M., Lasso M., Vidal P., Pandolfi D.; "Mantenimiento de locaciones petroleras mediante un Algortimo Multirecombinativo"; *8° Congreso Interamericano de Computación Aplicada a la Industria de Procesos (CAIP)*; Asunción-Paraguay, julio; (2007).

Selección de Centroides para Algoritmos de Clustering a través de Técnicas Metaheurísticas

Andrea Villagra, Daniel Pandolfi
Universidad Nacional de la Patagonia Austral
Ruta 3 Acceso Norte s/n
(9011) Caleta Olivia - Santa Cruz - Argentina
{avillagra,dpandolfi}@uaco.unpa.edu.ar

and

Guillermo Leguizamón
Universidad Nacional de San Luis,
Ejército de los Andes 950, (5700) San Luis, Argentina
legui@unsl.edu.ar

Abstract

The clustering algorithms like c-means are sensitive to the initialization values of the cluster centers and can be trapped by local extrema. In these terms, the use of estimated approaches to obtain the most appropriate cluster centers can be of great utility as a complementary tool during certain phases of the process of data mining; particulary, in some specific task of data mining, e.g., clustering.
In this way, Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) are two population metaheuristic approaches that could be considered as optimization. In this work the use of these two metaheuristic approaches is analyzed to optimize the initialization of the cluster centers values in the functions applied in the c-means algorithms. The respective results are compared using several datasets artificially generated.

keywords: Clustering, c-means, genetic algorithms, particle swarm optimization.

Resumen

Los algoritmos de *clustering* de tipo *c-means* son sensibles a los valores de inicialización de los centroides y pueden quedar atrapados en extremos locales. Planteado en estos términos, el uso de enfoques aproximados para obtener los centroides más adecuados puede ser de gran utilidad como herramienta complementaria durante ciertas fases del proceso de minería de datos, y en particular dentro de las tareas típicas de minería de datos, entre ellas la de *clustering* o agrupamiento. En esta dirección, los *Algoritmos Genéticos* (AGs) y la *Optimización Basada en Cúmulo de Partículas* (PSO)[1] son dos técnicas metaheurísticas poblacionales que podrían utilizarse en este ámbito, más aún cuando los problemas pueden ser planteados como de optimización.

En este trabajo se analiza el uso estas dos técnicas metaheurísticas para optimizar la inicialización de los valores de centroides en las funciones aplicadas en los algoritmos de *clustering* tipo *c-means*. Los respectivos resultados son comparados usando varios conjuntos de datos generados artificialmente.

Palabras claves: *Clustering*, *c-means*, algoritmos genéticos, optimización basada en cúmulo de partículas.

[1]Corresponde a las siglas en inglés para "Particle Swarm Optimization" las que son usadas convencionalmente.

1. INTRODUCCIÓN

La minería de datos constituye el núcleo del análisis inteligente de los datos y ha recibido un gran impulso en los últimos tiempos motivado por distintas causas: a) el desarrollo de algoritmos eficientes y robustos para el procesamiento de grandes volúmenes de datos, b) un poder computacional más barato que permite utilizar métodos computacionalmente intensivos, y c) las ventajas comerciales y científicas que han brindado este tipo de técnicas en las más diversas áreas. Entre las áreas donde han sido utilizadas exitosamente las técnicas de minería de datos podemos mencionar distintas aplicaciones financieras y bancarias, análisis de mercado, seguros y salud privada, educación, procesos industriales, medicina, biología, bioingeniería, telecomunicaciones, Internet, turismo, deportes, etc.

Es importante diferenciar en la minería de datos, el tipo de tareas que se suelen abordar y las técnicas utilizadas en cada caso. Como ejemplos de tareas generales se pueden mencionar el aprendizaje de conceptos, clasificación, categorización, regresión, agrupamiento (o *clustering*), correlaciones y análisis de asociación. Estas tareas pueden ser abordadas mediante distintos métodos o técnicas que suelen adaptarse mejor de acuerdo a la tarea sobre la cual se trabajará. Entre las técnicas más conocidas se puede mencionar el aprendizaje de reglas de clasificación, reglas de asociación, reglas relacionales, reglas difusas, árboles de decisión (y regresión), ecuaciones de regresión, redes neuronales, metaheurísticas, etc.

Un caso especial es el de las metaheurísticas, las que pueden jugar un papel bien definido en las distintas etapas del proceso del análisis inteligente de los datos en general y la minería de datos en particular [1] y [6]. El uso de metaheurísticas en estos casos puede realizarse en forma aislada o en combinación con otros algoritmos incluyendo el aprendizaje de redes neuronales y árboles de decisión. Por lo tanto, las metaheurísticas pueden ser usadas para mejorar la robustez y precisión de las técnicas más tradicionales usadas en extracción de características, selección de características, clasificación y agrupamiento (*clustering*). Éste último, un caso particular del aprendizaje no-supervisado, ha mostrado ser muy útil en el análisis exploratorio de datos, segmentación de imágenes y con algunas clases de conocimiento agregado, puede ser usado también para clasificación.

En este trabajo se presentan dos enfoques, uno guiado genéticamente y otro a través de cúmulo de partículas para optimizar un problema de *clustering*. Estos enfoques pueden ser directamente aplicados a cualquier modelo de *clustering* que pueda representarse como una función dependiente de un conjunto de centroides (centros de *cluster* o puntos prototipos). En un futuro pueden generalizarse para modelos que requieran parámetros en lugar de centroides. Ambos enfoques utilizan una represententación binaria de los centroides y la función (J_1) se usa como posible función objetivo dentro del enfoque *hard c-means* (denotado como HCM) tal como es usado en [9] y [12]. Una característica negativa de estos algoritmos de *clustering*, es que pueden quedar atrapados por extremos locales en el proceso de optimizar el criterio de *clustering*. Siendo además, muy sensibles a la inicialización de los centroides. En este trabajo se ha optado por dos metaheurísticas como herramientas alternativas para el problema planteado anteriormente. La primera, los Algoritmos Genéticos (AGs) introducidos por Holland en [5], inspirados en la capacidad de la naturaleza para evolucionar seres para adaptarlos a los cambios de su entorno. Los AG se han utilizado mucho en optimización y particularmente en problemas de *clustering* con resultados muy satisfactorios [11]. Por otro lado, se plantea el uso de una metheurística desarrollada más recientemente; los algoritmos Basados en Cúmulos de Partículas o *Particle Swarm Optimization* (PSO) [8], técnicas metaheurísticas inspiradas en el comportamiento social del vuelo de las bandadas, movimiento de los cardúmenes, entre otros sistemas sociales altamente cohesionados y que permite, simulando este modelo de comportamiento, obtener métodos eficientes para resolver problemas de optimización. En consecuencia, los enfoques presentados y analizados aquí son usados como una alternativa para la inicialización de los centroides con el fin de

aminorar la sensibilidad a la inicialización de los algoritmos de *clustering* de tipo *c-mean*.

El resto del trabajo está organizado como sigue. La siguiente sección ofrece una descripción general del denominado *hard c-means* clustering o HCM y la descripción del AG y PSO para su resolución. La sección 3 muestra los resultados y el análisis de los experimentos realizados; finalmente, son presentadas las conclusiones y posibles direcciones para estudios futuros.

2. CLUSTERING CON HCM

En esta sección se describe el enfoque HCM sobre el cual se aplican las metaheurísticas para encontrar los mejores centroides iniciales. Consideremos un conjunto de n vectores $X = \{x_1, x_2, \ldots, x_n\}$ a ser agrupados en c grupos de datos parecidos. Cada x_i es un vector de características del objeto representado por x_i. Las características pueden ser longitud, ancho, color, etc.

La forma en que se determina la pertenencia de un objeto a un *cluster* u otro se puede determinar, por ejemplo, de manera estricta (llamado *hard*) o por el contrario, de manera más relajada (llamado *fuzzy*) en la cual un objeto puede pertenecer a diferentes *clusters*, pero con diferentes niveles de pertenencia. En este trabajo, estamos interesados en el primer caso, o *hard clustering*. De aquí en más asumiremos este tipo de agrupamiento estricto o *hard* cuando hagamos referencia a un *cluster*.

Los distintos *clusters* de los objetos pueden ser representados por una matriz de miembros llamada partición *hard*. El conjunto de datos $c \times n$ matrices de partición *hard* no degeneradas se denota por M_{cn} y se define como:

$$M_{cn} = \left\{ U \in R^{c \times n} \middle| \sum_{i=1}^{c} U_{ik} = 1, 0 < \sum_{k=1}^{n} U_{ik} < n, \text{ y } U_{ik} \in \{0,1\}; 1 \leq i \leq c; 1 \leq k \leq n. \right\} \quad (1)$$

Mientras que el criterio para determinar la calidad de un *cluster* para particiones *c-means* puede ser medido por la función HCM:

$$J_1(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (U_{ik}) D_{ik}^2(v_i, v_k) \quad (2)$$

donde $U \in M_{cn}$ es una matriz de partición *hard*; $V = [v_1, ..., v_c]$ es una matriz de parámetros centroides $v_i \in R^s$ y $D_{ik}(v_i, v_k)$ es una medida de distancia de v_k al i-ésimo centroide. Por ejemplo, en [9] se usa como métrica de distancia la Euclideana.

2.1. Algoritmos Genéticos

Los algoritmos genéticos fueron desarrollados por John H. Holland a principios de los 1960s [3], [4], motivado por resolver problemas de aprendizaje de máquina. Desde entonces, los AGs han evolucionado rápidamente (junto con otros algoritmos de la familia de algoritmos evolutivos) hasta llegar a conformar una de las metaheurísticas más ampliantente estudiadas y aplicadas durante los últimos años ([15], [2], [13], [14]).

En términos generales, el AG enfatiza la importancia de la recombinación (operador principal) sobre el de la mutación (operador secundario), y usa selección probabilística.
Los pasos del algoritmo genético básico descripto en forma general son los siguientes:

- Generar (aleatoriamente) una población inicial.
- Calcular la aptitud de cada individuo.
- Seleccionar (probabilísticamente) en base a la aptitud.
- Aplicar operadores genéticos (recombinación y mutación) para generar la siguiente población.
- Ciclar hasta que cierta condición se satisfaga.

Para el problema de *clustering* según planteado al inicio de esta sección, se puede visualizar a la población del AG de la siguiente manera: en cualquier generación el elemento i de una población es V_i, una matriz de $c \times n$ centroides en la notación HCM. La población inicial de tamaño P se contruye por la asignación aleatoria de números para cada una de las s características de los c centroides. Los valores iniciales están restringidos a estar en el rango (determinado por el conjunto de datos) del cual son asignados, pero por otra parte son aleatorios. Debido a que únicamente los Vs serán usados por el AG es necesario reformular la función objetivo (2) para la optimización. Para cada vector de datos HCM se asigna el *cluster* más cercano a través de la distancia métrica (en este caso se usa la distancia Euclideana, aunque otras alternativas son posibles).

Dada la manera en que las asignaciones a los *cluster* se hacen, se da que

$$R_1(V) = \sum_{k=1}^{n} \text{mín}\{D_{1k}, D_{2k}, \ldots, D_{ck}\} \tag{3}$$

es una reformulación equivalente a J_1 que elimina U [11].

En el Algoritmo 1 se muestra el algoritmo genético simple utilizado en este problema de *clustering*. El algoritmo crea una población inicial $P(0)$ con μ soluciones, en este caso con c centroides y luego evalua esas soluciones. La evaluación se realiza utilizando la ecuación 3. Luego la población ingresa en un ciclo donde evoluciona, lo cual significa que se le aplica operadores de recombinación y mutación y se crean λ hijos. Finalmente, cada iteracion finaliza seleccionando μ individuos para construir la nueva población. En este caso, el criterio de parada para el ciclo es alcanzar el número máximo de evaluaciones ($maxEvaluaciones$).

Algorithm 1 Algortimo Genético Simple

t=0; {generación actual}
inicializa $(P(t))$;
evalua $(P(t))$;
while (**not** $maxEvaluaciones$) **do**
  $P'(t) =$ evoluciona $(P(t)$; {recombinación y mutación}
  evalua$(P'(t))$;
  $P(t+1) =$ selecciona la nueva población de $P'(t) \bigcup P(t)$
  $t = t + 1$;
end while

2.2. Optimización Basada en Cúmulo de Partículas

El PSO es un algoritmo que utiliza una población de soluciones potenciales que evolucionan a la solución óptima (o muy cercana de ésta) de un determinado problema. La principal diferencia del

PSO y la computación evolutiva es un mecanismo que no parece tener un análogo en estas últimas, en el cual las partículas o individuos sobrevuelan a través del hiperespacio de búsqueda del problema [8]. El movimiento de una partícula está influenciado por su velocidad y las posiciones donde se encontraron buenas soluciones.

Para el problema de *clustering* estudiado en el presente trabajo se utilizó un modelo de PSO binario donde cada individuo de la población sólo tiene en mente la decisión binaria que tiene que tomar: si/no, falso/verdadero, etc. Cada individuo está rodeado por otros que también tienen que tomar su propia decisión. La información que tiene disponible para poder tomar la mejor decisión es su propia experiencia y la experiencia de sus vecinos. Para ello, Kennedy y Eberhart [7] proponen un modelo matemático donde la probabilidad de que un individuo decida si/no, falso/verdadero o alguna otra decisión binaria es una función f que depende de factores personales y sociales:

$$P[x_{id}(t) = 1] = f[x_{id}(t-1), v_{id}(t-1), p_{id}, p_{gd}] \tag{4}$$

donde:

- $P[x_{id}(t) = 1]$ es la probabilidad de que el individuo i seleccione 1 para el bit en la posición d de la cadena binaria.
- $x_{id}(t)$ es el estado actual del bit d de la cadena binaria
- $v_{id}(t-1)$ es la medida de la predisposición individual o la probabilidad actual de seleccionar 1.
- p_{id} es el mejor estado encontrado para el bit d de la cadena binaria de acuerdo a la experiencia personal del individuo.
- p_{gd} es el mejor estado encontrado para el bit d de la cadena binaria encontrado por el mejor individuo del vecindario.

Si $v_{id}(t)$ es grande, el individuo tiene una mayor predisposición a seleccionar 1, mientras que valores pequeños favorecen al 0. Para ubicar el valor de este parámetro dentro del rango $[0{,}0; 1{,}0]$, se utiliza la siguiente función sigmoidal:

$$s(v_{id}) = \frac{1}{1 + \exp(-v_{id})} \tag{5}$$

Para ajustar la disposición de cada individuo hacia el éxito personal y de la comunidad, se construye una fórmula para cada v_{id} en el momento actual, que será una función de la diferencia entre el estado actual del individuo y el mejor estado encontrado por él y sus vecinos. En cualquier situación no se puede determinar cuál será el factor de mayor influencia para tomar la decisión: el factor individual o el social, por lo cual cada uno de estos factores es multiplicado por un número aleatorio para que se vaya alternando el efecto de cada uno de ellos. La fórmula de decisión binaria es:

$$\begin{gathered} v_{id} = v_{id}(t-1) + \phi_1 r_1[p_{id} - x_{id}(t-1)] + \phi_2 r_2[p_{gd} - x_{id}(t-1)] \\ \text{Si } w_{id} < s[v_{id}(t)] \text{ entonces } x_{id}(t) = 1; \text{ en otro caso, } x_{id}(t) = 0 \end{gathered} \tag{6}$$

donde ϕ_1 y ϕ_2 representan constantes positivas de aceleración que escalan respectivamente la contribución cognitiva y social; $r_1, r_2 \sim U(0,1)$ y w_{id} es un vector de números aleatorios entre 0,0 y 1,0 con distribución uniforme. Cada una de estas fórmulas es aplicada repetidamente en cada una de las dimensiones de cada individuo, verificando cada vez si el valor actual de x_{id} resulta en una mejor evaluación que p_{id}, en cuyo caso se actualiza el valor. Algunas veces la decisión que tome el individuo estará basada en su experiencia personal y otras en su percepción de lo que los otros individuos creen. El version binaria de PSO binario se muestra en el Algoritmo 2.

Algorithm 2 Versión Binaria de PSO

repeat
  for cada individuo i en la población **do**
    if $aptitud(\overrightarrow{x_i}) > aptitud(\overrightarrow{p_i})$
      for cada dimensión d del individuo **do** $p_{id} = x_{id}$
    $g = i$
    for cada vecino j del individuo i **do**
      if $aptitud(\overrightarrow{p_j}) > aptitud(\overrightarrow{p_g})$
        $g = j$
    for cada dimensión d del individuo **do**
      $v_{id} = v_{id}(t-1) + \phi_1 r_1[p_{id} - x_{id}(t-1)] + \phi_2 r_2[p_{gd} - x_{id}(t-1)]$
      $v_{id} \in (-V_{max}, V_{max})$
      if $w_{id} < s[v_{id}(t)]$
        $x_{id}(t) = 1$
      else $x_{id}(t) = 0$
  end for
until Alcance condición de parada

Para el algoritmo PSO existe una gran variedad de parámetros que pueden ser ajustados para modificar la manera en que éste lleva a cabo la búsqueda. Los dos parámetros más importantes son V_{max} y ϕ los cuales son establecidos al inicio del algoritmo y se utilizan a lo largo de toda la búsqueda. La manipulación de estos parámetros puede causar cambios abruptos en el comportamiento del sistema. El parámetro V_{max} es utilizado para evitar que la trayectoria de la partícula se salga de control y que se expanda en ciclos cada vez más amplios en el espacio del problema hasta que eventualmente tienda al infinito. El parámetro de control ϕ también es llamado la constante de aceleración y determina el tipo de trayectoria que tomarán las partículas. La aptitud de una partícula se calcula de la misma forma que se calculó para el algoritmo AG, utilizando la ecuación 3.

3. EXPERIMENTOS Y RESULTADOS

Para probar ambos enfoques metaheurísticos estudiados en este trabajo (es decir, AGs y PSO) se generó un conjunto de datos artificiales que tiene múltiples extremos locales. Dicho conjunto de objetos tiene dos características vinculadas entre sí por la salida y de la ecuación no lineal $y = (1 + x_1^{-2} + x_2^{-1,5})^2$, donde $1 \leq x_1, x_2 \leq 5$. Esta ecuación ha sido tomada de [10], donde se encontraron seis clases para proveer un agrupamiento útil en el desarrollo de reglas difusas para modelar la salida de la ecuación sobre un conjunto de 50 valores distintos. Para el análisis realizado en el presente trabajo se generaron 10 instancias de 50 valores cada una. Para cada enfoque metaheurístico se mostrarán dos

experimentos diferentes cambiando ciertos parámetros de cada una de las metaheurísticas estudiadas aquí. Cada experimento consistió en la ejecución de 30 corridas independientes para cada una de las 10 instancias consideradas.

3.1. Asignación de valores de parámetros

Para analizar la eficiencia de los algoritmos se ha realizado un estudio previo de parámetros y se decidió mostrar los resultados obtenidos utilizando un tamaño de población (cúmulo de partículas) grande en el primer caso y pequeño en el segundo caso, con la finalidad de observar la calidad de los resultados en cada caso.

En ambos enfoques se utilizó una representación binaria del individuo o partícula, según corresponda. Donde un individuo representa una solución que contiene los seis posibles centroides. Por lo tanto, cada individuo se representó por medio de un arreglo de 96 elementos binarios. De los cuales se usaron 8 bits para representar un valor de x_1 o de x_2 con una precisión de dos lugares decimales. Como en un individuo se representaron 6 centroides, esto hizo un total de 96 $(16 * 6)$ elementos por individuo. En la Figura 1 se muestra la representación de un individuo o partícula según corresponda.

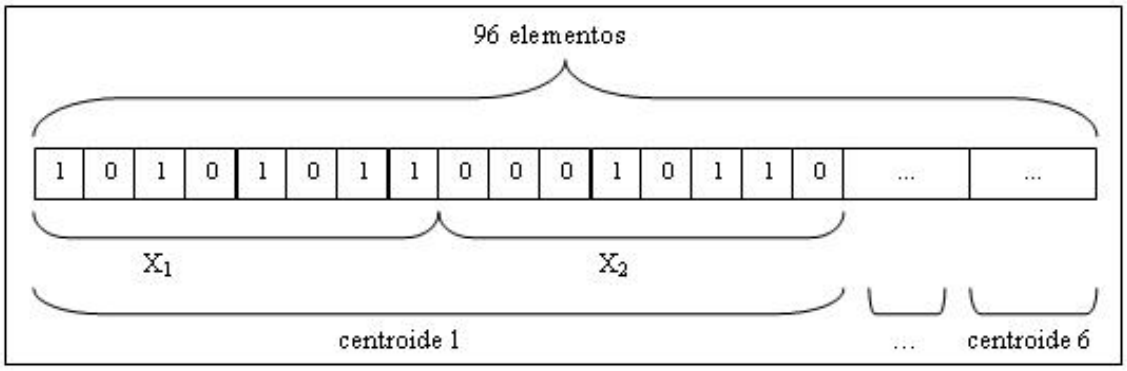


Figura 1: Representación de una solución

Para el AG se fijó una población de 250 individuos, y el número máximo de generaciones se fijó en 5000 para el primer experimento; mientras que para el segundo, se fijó una población de 25 individuos y el número máximo de generaciones de 1000. En ambos casos se aplicó un operador de recombinación de dos puntos o *Two Point Crossover* (TPX) con una probabilidad de 0,65, y el operador de mutación que se utilizó fue el de intercambio de un bit o *Bit-Flip mutation* con una probabilidad de 0,05.

Para el PSO en el primer caso se fijó el cúmulo de partículas en 600 partículas y el máximo de iteraciones en 500 y para el segundo caso se fijó el cúmulo de partículas en 10 partículas y el máximo de iteraciones en 1000. En ambos casos las constantes ϕ_1, ϕ_2 se fijaron en 2 respectivamente. El tamaño del vecindario se estableció igual a toda la población, usándose una versión global del PSO. En la Tabla 1 se muestran los parámetros utilizados por los algoritmos en cada caso planteado anteriormente, es decir, AG1, AG2, PSO1 y PSO2, respectivamente.

En la Tabla 2 se presentan los resultados obtenidos con el AG1 y PSO1 utilizando paramétros definidos para cada uno de ellos. La primer columna representa el número de la instancia, las columnas 2 y 3 corresponden a los valores mínimos (Min) y máximos (Max) obtenidos por el AG en las 30 corridas efectuadas, la columna 4 corresponde al promedio (MedEval) de evaluaciones requeridas por

Tabla 1: Parámetros de los algoritmos AG y PSO, donde NC indica “No Corresponde”.

Parámetros	AG1	AG2	Parámetros	PSO1	PSO2
Tamaño Población	250	25	Tamaño Cúmulo	600	10
Número Máx. de Generaciones	5000	1000	Número Máx. de Iteraciones	500	1000
Recombinación	TPX	TPX	ϕ_1	2	2
Mutación	Bit-Flip	Bit-Flip	ϕ_2	2	2
Prob. Recombinación	0,65	0,65	NC	NC	NC
Prob. Mutación	0,05	0,05	NC	NC	NC

dicho algortimo para encontrar el valor mínimo. Similarmente, las columnas 5 y 6 corresponden a los valores mínimos y máximos obtenidos por el PSO en las 30 corridas efectuadas y la ultima columna (columna 7) corresponde al número de evaluaciones promedio necesarias por PSO para encontrar el valor mínimo. Se puede observar que para ambos enfoques los valores mínimos y máximos de la función objetivo difieren muy poco, en todos los casos los valores obtenidos son bastante similares (las posibles diferencias entre ellos no son estadísticamente significativos acorde al análisis de Kruskawallis a un nivel de confianza del 95 %). Sin embargo, existe una importante diferencia en el promedio de evaluaciones (Med.Eval) necesarias en cada enfoque para alcanzar el resultado. Para AG1 el número de evaluaciones promedio varia entre 532225 (intancia 9) y 781883 (instancia 2) mientras que para el PSO1 el número de evaluaciones promedio varia entre 52700 (instancia 8) y 104220 (instancia 10).

Tabla 2: Resultados obtenidos con AG1 y PSO1

	AG1			PSO1		
Instancia	Min	Max	MedEval	Min	Max	MedEval
1	0,0904	0,0947	685283	0,0902	0,0916	77080
2	0,0892	0,0922	781883	0,0890	0,0905	68100
3	0,1043	0,1126	675142	0,1040	0,1123	69120
4	0,1463	0,1588	681333	0,1460	0,1546	81520
5	0,1165	0,1234	610658	0,1165	0,1166	55640
6	0,1195	0,1237	588558	0,1195	0,1202	52620
7	0,1069	0,1262	555492	0,1068	0,1076	67660
8	0,1353	0,1418	533083	0,1352	0,1363	52700
9	0,1098	0,1355	532225	0,1095	0,1104	65440
10	0,0941	0,0962	587425	0,0939	0,0946	104220

En la Tabla 3 se muestra el mismo tipo de información que en la Tabla 2 con resultados obtenidos por los algoritmos AG2 y PSO2. Se puede observar nuevamente que para ambos enfoques los valores mínimos y máximos de la función objetivo difieren muy poco, en todos los casos los valores obtenidos son bastante similares (las posibles diferencias entre ellos no son estadísticamente significativos acorde al análisis de Kruskawallis a un nivel de confianza del 95 %). Aunque igual al caso de AG1 y PSO1, existe una importante diferencia en el promedio de evaluaciones necesarias en cada enfoque para alcanzar los mejores valores. Así, para AG2 el número de evaluaciones promedio varía

entre 6667 (intancia 7) y 12899 (instancia 8), mientras que para el PSO2 el número de evaluaciones promedio varía entre 1682 (instancia 10) y 2560 (instancia 7).

Tabla 3: Resultados obtenidos con AG2 y PSO2

	AG2			PSO2		
Instancia	Min	Max	MedEval	Min	Max	MedEval
1	0,0941	0,1136	10979	0,0904	0,0959	1793
2	0,0933	0,1568	9540	0,0890	0,0928	2040
3	0,1135	0,1585	12155	0,1042	0,1438	1704
4	0,1588	0,1893	8325	0,1468	0,1577	1798
5	0,1199	0,1433	9666	0,1165	0,1219	2198
6	0,1224	0,1646	10473	0,1195	0,1428	1770
7	0,1158	0,1715	6667	0,1069	0,1268	2560
8	0,1402	0,1906	12899	0,1352	0,1391	2202
9	0,1194	0,1860	12284	0,1098	0,1427	1781
10	0,0969	0,1130	10830	0,0940	0,1005	1682

En la Figura 2 se muestra las curvas de nivel de la función utilizada en este problema $y = (1 + x_1^{-2} + x_2^{-1,5})^2$, donde $1 \leq x_1, x_2 \leq 5$), los 50 puntos (símbolo " + ") correspondientes a la instancia 2 y los seis centoides (símbolo " ∘ ") encontrados por el algoritmo AG2 para esa instancia.

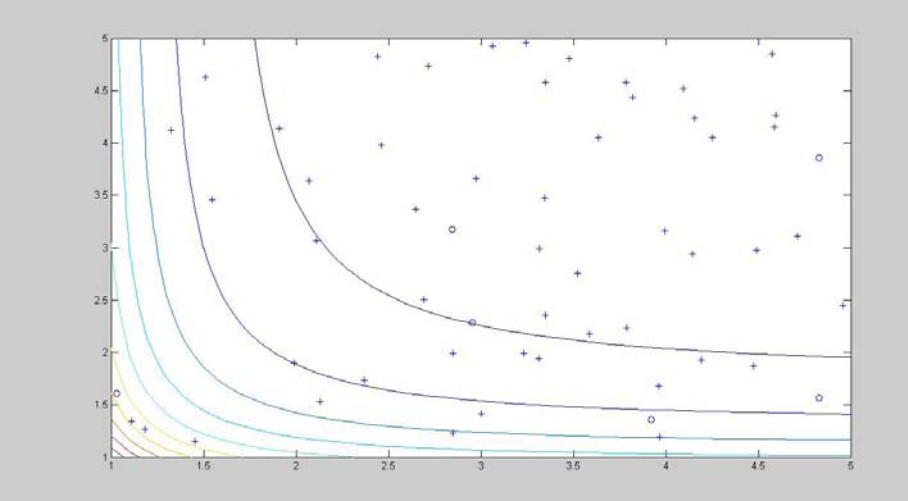


Figura 2: Distribución de centroides (AG2)

En la Figura 3 se muestra las curvas de nivel de la función utilizada en este problema $y = (1 + x_1^{-2} + x_2^{-1,5})^2$, donde $1 \leq x_1, x_2 \leq 5$), los 50 puntos (símbolo " + ") correspondientes a la instancia 2 y los seis centoides (símbolo " ∘ ") encontrados por el algoritmo PSO2 para esa instancia.

4. CONCLUSIONES

En este trabajo se ha presentado el uso de dos metaheurísticas AG y PSO para aminorar el problema de elección de los valores de inicialización para los algoritmos de *clustering c-means*. Se ha

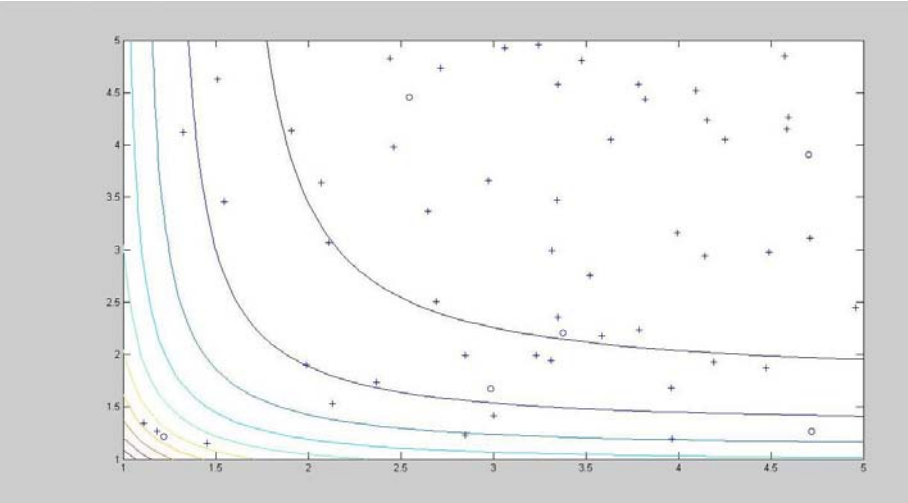


Figura 3: Distribución de centroides (PSO2)

realizado una serie de experimentos preliminares utilizando una ecuación no lineal de una sóla característica que contiene múltiples extremos locales. Los resultados demuestran que ambas técnicas proveen centroides bastante similares, pero PSO requiere un menor número de evaluaciones para obtener los resultados. Para un tamaño de población grande PSO requiere un 86 % menos en el número de evaluaciones promedio respecto al AG. Por otro lado, cuando el tamaño de la población es menor, el número de evaluaciones promedio del PSO se reduce en un 80 % respecto al AG. Consecuentemente, se puede concluir que independientemente del tamaño de la población PSO obtiene resultados muy similares con una importante reducción del número de evaluaciones requeridas.

Como trabajo futuro se pretende analizar en detalle el porque de la disminución abrupta en el número de evaluaciones del algoritmo PSO respecto al AG, lo que se produjo sin comprometer la calidad de los resultados. Además, se incluirá con un estudio en profundidad considerando instancias con una mayor cantidad de características y de mayor cantidad de datos y objetos a agrupar.

5. AGRADECIMIENTOS

EL primer y segundo autor agradecen a la Universidad Nacional de la Patagonia Austral por su apoyo al grupo de investigación y además, la cooperación de los integrantes del proyecto que continuamente proveen de nuevas ideas y críticas constructivas. El tercer autor agradece el constante apoyo brindado por la Universidad Nacional de San Luis y la ANPYCIT que financian sus actuales investigaciones

REFERENCIAS

[1] Hussein A., Ruhul A. S., and Charles S. N. *DATA MINING: A heuristic approach*. Idea Group Publishing, 2002.

[2] Goldberg D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley Longman, Inc, 2002.

[3] Holland J. H. Concerning efficient adaptive systems. In M.C. Yovits, G.T. Jacobi, and G.D. Goldstein, editors, *Self-Organizing Systems-1962*, pages 215–230, Washington D. C., 1962. Spartan Books.

[4] Holland J. H. Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery*, pages 9:297–314, 1962.

[5] Holland J. H. *Adaptation in Natural and Artificial Systems*. The MIT Press, Cambridge, Massachusetts, first edition edition, 1975.

[6] Witten I. H. and Frank E. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[7] Kennedy J. and Eberhart R. A discrete binary version of the particle swarm algorithm. In *Proceedings of the 1997 IEEE Conference on Systems, Man, and Cybernetics*, pages 4104–4109, Piscataway, New Jersey, 1997. IEEE Service Center, 1997.

[8] Kennedy J. and Eberhart R. *Swarm Intelligence*. Morgan Kaufmann, San Francisco, California, 2001.

[9] Bezdek J.C. *Pattern Recognition with Fuzzy Objective Functions*. 1981.

[10] Sugeno M. and Yasukawa T. A fuzzy logic based approach to qualitative modeling. *IEEE Trans. Fuzzy Systems*, 1:7–31, Feb 1993.

[11] Hall L. O., Özyurt I. B., and Bezdek J. C. Clustering with a genetically optimized approach. In *IEEE Transactions on Evolutionary Computation*, volume 2, pages 103–112, July 1999.

[12] Duda R.O. and Hart P.E. *Pattern Classification and Scence Analysis*. Wiley, new york edition, 1973.

[13] Bäck T. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[14] Bäck T., Fogel D.B., Whitley D., and Angeline P.J. Mutation operators. In *Evolutionary Computation 1. Basic Algorithms and Operators*. IOP Publishing Lt., 2000.

[15] Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1996.

Sistema de Asistencia Ventricular Izquierda: Arquitectura y Sistema para el Monitoreo a Distancia de la Unidad Electroneumática.

Cervino, Claudio[1], **Jorge Ierache**[1, 2], **Víctor Nasini** [1]
[1] Instituto de Investigaciones Básicas y Clínicas en Enfermedades Cardiovasculares. Facultad de Medicina, Universidad de Morón.
[2] Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica ISIER
Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales. Universidad de Morón.
Cabildo 134, (B1708JPD) Morón, Buenos Aires, Argentina.
Tel: 54-11-5627-2000 - Fax: 54-11-5627-2002
ccervino@unimoron.edu.ar jierache@unimoron.edu.ar vnasini@unimoron.edu.ar

Abstract

The development of an electropneumatic unit remote monitoring system for the Left Ventricular Assistance appears in the context of telemedicine, the particular results in relation to the development of the remote monitoring system, and the patient´s physiological parameters it monitors: ECG, arterial pressure, corporal temperature, electropneumatic unit operation, and among these ones the programmed values of pneumatic pressure, compressor temperature, and level of synchronization. On the other hand the work involves the characteristics of an architecture for the dynamic integration of the patient´s monitoring.

Key words: Monitoring of remote patients, Applications, Telemedicine, Left Ventricular Assistance System, System Architectures, Bioinformatics

Resumen

Se presenta en el contexto de la telemedicina el desarrollo de un sistema de monitoreo a distancia de una unidad electroneumática para la Asistencia Ventricular Izquierda, se exploran los resultados particulares con relación al desarrollo del sistema de monitoreo remoto, y los parámetros fisiológicos del paciente que el sistema monitorea: ECG, presión arterial, temperatura corporal, funcionamiento de la unidad electroneumática, entre estos los valores programados de presión neumática, temperatura de los compresores, nivel de sincronización. En otro orden el trabajo aporta las características de una arquitectura para la integración dinámica del monitoreo de pacientes.

Palabras clave: Monitoreo de pacientes a distancia, Aplicaciones móviles, Telemedicina, Sistema de Asistencia Ventricular Izquierda, Arquitecturas de Sistemas, Bioinformática.

1. INTRODUCCION

Las funciones que la Telemedicina podía ofrecer en sus inicios, en los años 70, eran muy limitadas y comúnmente a los pacientes les servia para consultar información médica [3]. Hoy en día, el significado de telemedicina se relaciona con el uso tecnológico de los sistemas de información proveyendo información médica y servicios para miles de propósitos, como diagnósticos de enfermedades, transferencia de datos y registros médicos, monitoreo de rehabilitación de pacientes o procesos de terapias. Además, en los últimos tiempos la tradicional plataforma de telemedicina de escritorio ha evolucionado a plataformas más prácticas, como las plataformas de telemedicina wireless[1] y dispositivos móviles que faciliten la monitorización a distancia de señales biológicas vitales a través de radio [10], [11].

La Organización Mundial de la Salud define la Telemedicina como la distribución de servicios de salud, en el que la distancia es un factor crítico, donde los profesionales de la salud usan información y tecnología de comunicaciones para el intercambio de información válida para el diagnóstico, tratamiento y prevención de enfermedades o daños, investigación y evaluación; y para la educación continuada de los proveedores de salud pública, todo ello en interés del desarrollo de la salud del individuo y su comunidad [14]. La aplicación de la telemedicina a los sistemas médicos puede lograr una mayor eficacia de los servicios de salud y un mejor acceso a la atención de personas con discapacidades o ancianos [1].

La telemedicina se refiere a todas las formas que posee el intercambio de información médica, incluyendo una variedad de tecnologías de comunicación. Las aplicaciones referidas a la medicina y al cuidado de la salud incluyen telecomunicaciones, y datos que son usados para transferir información médica [2]. En este contexto la Telemedicina puede ser definida como la transmisión de información médica y la prestación de servicios de salud a través de redes de telecomunicaciones. Esta incluye la transmisión de imágenes fijas, vídeo y otras formas de datos médicos. El principal objetivo de la telemedicina es que el usuario de los datos monitorizados pueda tener un acceso integral a la información en cualquier momento, desde cualquier lugar y en la mejor forma posible adecuada a las capacidades de ancho de banda y dispositivos disponibles. Otro objetivo de los servicios de telemedicina es el desarrollo y evaluación de servicios de monitoreo y telecuidado para pacientes en su lugar de internación o en su propio domicilio, la registración de datos para predicción y empleo de sistemas expertos e inteligente de datos. Entre los factores determinantes en la transmisión de datos médicos están la velocidad, el ancho de banda y el retardo en las transmisiones que algunas veces llega a ser crítico en las aplicaciones de la telemedicina [6].

Los usuarios del sistema, médicos, enfermeros y demás profesionales de la salud, podrán hacer uso de esta tecnología desde la misma red de área local, dentro del hospital donde esté instalado, como también desde puntos remotos a través de Internet, con la posibilidad de usar distintos tipos de aplicaciones cliente. En la actualidad existen varios proyectos *open source*[2] sobre telemedicina en varias plataformas diferentes, pero no ocurre lo mismo a la hora de buscar proyectos basados en el Framework de .NET. Entre los proyectos de telemedicina *open source* vivientes y relevantes podemos mencionar el proyecto iPath [5] de la universidad de Basel que consta de una plataforma de colaboración para el intercambio de conocimientos médicos y de enseñanza médica a distancia; así como también, el portal Open ECG [12] que además de publicar varios sistemas *open source*, publica estándares abiertos sobre ECG y la telemedicina. Otro caso es el de PhysioNet [13] proyecto del MIT que incluye una gran biblioteca de señales biomédicas

[1] Wireless: Referido a las telecomunicaciones, se aplica el término inalámbrico al tipo de comunicación en la que no se utiliza un medio de propagación físico, sino se utiliza la modulación de ondas electromagnéticas, las cuales se propagan por el espacio sin un medio físico que comunique cada uno de los extremos de la transmisión

[2] Código abierto (del inglés *open source*) es el término por el que se conoce al software distribuido y desarrollado en forma libre. Este término empezó a utilizarse en 1998 por algunos usuarios de la comunidad del software libre.

capturadas de varios hospitales para su intercambio, el análisis cooperativo de los datos obtenidos, la evaluación nuevos algoritmos propuestos a partir del análisis y el desarrollo del software *open source*, consecuente de la aplicación de estos algoritmos. El acceso a las redes en telemedicina se realiza por diferentes vías: Intranet, Internet, etc. Las señales de ECG, PA, oximetría, etc. requieren de un proceso para su posterior análisis y almacenamiento. En dichas variables, se puede llevar a cabo la estimación en tiempo real de las mismas o se puede realizar la compresión y almacenamiento masivo en bases de datos.

Los Sistemas de Asistencia Ventricular Izquierda (SAVI) se han desarrollado durante las últimas cuatro décadas [7], [8], [4]. Su principal función es dar apoyo al corazón frente a la insuficiencia cardíaca. Desde 1967, el SAVI es utilizado como puente hasta que el corazón de un donante esté disponible para el trasplante [9]. El SAVI conecta al corazón del paciente con el resto de la circulación y proporciona apoyo circulatorio tomando parte del trabajo del corazón enfermo. El flujo de sangre se dirige desde el corazón izquierdo hacia la bomba sanguínea. Luego, el ventrículo artificial eyecta sangre a través de una cánula de salida, por la aorta, al sistema arterial del paciente. El sistema desarrollado en la Facultad de Medicina de la UM es una bomba sanguínea neumáticamente impulsada. El dispositivo implantable está unido externamente a una Unidad Electroneumática (*driver*). La conexión neumática percutánea ingresa por la pared abdominal, y transmite variaciones de presión neumática -compresión de aire y vacío-. El pulso de aire comprime a la bolsa sanguínea del ventrículo artificial, obligando a la sangre a través de válvulas a ingresar al sistema arterial [9], [4].

En el presente trabajo se propone el desarrollo de un sistema de monitoreo a distancia para ECG, PA y otros parámetros del paciente y de diversos parámetros de funcionamiento del driver.

2. ESTADO ACTUAL DEL SISTEMA

La Unidad Electroneumática (UEN) genera la compresión del aire y el vacío, que es transportado a través del tubo neumático, y de esta manera acciona a la bomba sanguínea intracorporea. El largo del tubo neumático desde el paciente a la UEN tiene una longitud de 2,5 metros. Se busca que sea lo suficientemente largo para que el paciente tenga una movilidad cómoda en la cama durante el reposo, pueda movilizarse por el hospital o en su domicilio con una cierta libertad y además pueda proceder al aseo personal adecuadamente sin que se corra el riesgo de afectar a la UEN**.**

La *UEN* está formada por tres Módulos: (1) Control Principal (MCP), (2) Potencia (MP), y (3) Cargador de Baterías**.** El *driver* posee un importante Sistema de Seguridad y Alarmas. A su vez, desde el MCP puede derivarse una salida (Monitor Output) hacia un Módulo de Monitoreo (interfase, computadora y programa). La UEN puede funcionar en 4 (cuatro) Modos distintos: (1) Modo Fijo Manual,(2) Modo Automático (Fijo Secundario),(3) Modo Programado y Modo Sincrónico. Clínicamente, el *SAVI* debe utilizarse en el Modo Programado ó Modo Sincrónico.

La UEN está diseñada de tal forma que posee sistemas electrónicos y neumáticos gemelos, denominándose a cada conjunto como GRUPO. Los mismos trabajan en forma alternada cada 10 minutos, esta duplicación del grupo neumático tiene como finalidad aumentar la vida útil de los compresores y válvulas electroneumáticas, ya que la UEN debe funcionar en forma permanente. Si estuviera formado por un solo juego compresor/electroválvula, el funcionamiento continuo de éstos se traduciría en un desgaste mecánico prematuro, y la temperatura de trabajo se elevaría a valores críticos, con lo que la confiabilidad del driver disminuiría. Al disponer de grupos idénticos que trabajan en forma alternada, es decir, durante un período de 15 min, un grupo permanece en marcha y el otro detenido, expirado dicho tiempo, los roles de los sistemas se invierten automáticamente.

En caso que uno de ellos fallase, se activará una alarma sonora y visual que indica el mal funcionamiento del mismo, pasándose el control al otro grupo que actuará en forma permanente, tomando la potencia del SAVI hasta su solución técnica. Nuevamente, este sistema dual permite garantizar un alto grado de confiabilidad, como así también confianza y tranquilidad en el paciente y personas que lo rodean. La UEN es una unidad que funciona con baterías (CC 12V). Dichas baterías son continuamente recargadas desde una conexión a la red eléctrica (CA 220V-50 Hz). Las baterías cargadas pueden mantener una autonomía del driver durante un período mínimo de 4 hs.

Se muestra en la. Figura 1. los módulos del sistema UEN UH2 a través del diagrama de bloques que presenta la disposición interna y sus relaciones. El Módulo de Potencia contiene al Sistema Neumático.

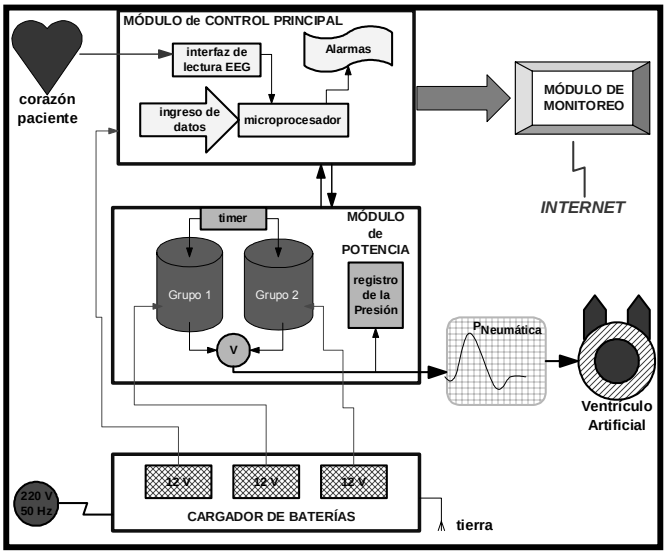


Figura 1. Módulos del sistema UEN UH2.

El Módulo de Monitoreo no está incorporado al driver. La UEN está compuesta por un gabinete que contiene al *Módulo de Control Principal (MCP)* y al *Módulo de Potencia (MP)*. Por debajo se encuentra un *Cargador de Baterías*. Ambos se hallan fijos a una estructura metálica con ruedas para su fácil ubicación y traslado. En el gabinete, todos los controles y el ingreso de señal ECG desde el paciente se encuentran en el *panel frontal* (parte superior). La salida neumática para la conexión del tubo neumático que alimenta a la BS sale del costado derecho. El funcionamiento de la UEN permite bombear aire comprimido por medio de un *Sistema Neumático (SN)*, que se encuentra dentro del MP, y que está constituido por compresores, válvulas y tanques reservorios (Fig. 1). El SN está comandado por circuitos electrónicos que controlan, los siguientes parámetros:

2.1 Frecuencia de bombeo y duración del período de inyección de aire (Período Sistólico): Puede ser controlado por: (1) el MCP, donde la fundamental característica, dentro de límites determinados, es la *sincronización* del bombeo de la BS con el ECG del paciente, o puede proceder (2) de un circuito electrónico de *Control Secundario (CS)*, que se encuentra dentro del MP.

2.2 Presión neumática: el control se realiza en forma independiente de los parámetros anteriores, y permite regular la presión neumática (positiva y negativa) de actuación del SAVI. Así, el SN suministra, con una determinada presión y frecuencia, aire comprimido y vacío alternativamente hacia la BS a través del tubo neumático.

3. DESCRIPCIÓN DE LOS MÓDULOS Y COMPONENTES DEL DRIVER

El MCP posee un "cerebro", que está formado por un Microcontrolador de la serie 8751/2 de Intel. Este microprocesador debe ser alimentado con determinados parámetros numéricos ingresados desde un Teclado (Data Entry), los cuales se visualizan a través de un Display (indicador numérico rojo).Los parámetros a ingresar anteriormente mencionados son los siguientes (en orden de ingreso) (Tabla I). La UEN tiene 4 (cuatro) modos de funcionamiento: Fijo Externo, Automático (Fijo Secundario), Programado, Sincrónico. En el modo de funcionamiento Modo Sincronizado, para que la BS trabaje en forma sincrónica con el corazón nativo, una Interfaz de Lectura de ECG electrónica (desde el MCP) toma la señal de ECG del paciente, calcula el período R-R y entrega una orden eléctrica de control al SN (del MP), para que éste trabaje en sincronismo con dicho ECG. Si el ECG del paciente se encuentra fuera del *rango de sincronización ó el corazón del paciente se torna arrítmico o fibrila*, un circuito del MP toma el control del SN y, según la frecuencia nominal programada, la BS bombea sangre en forma asincrónica.

Tabla I. Parámetros que deben ser ingresados por teclado que permiten programar a la UEN.

	Parámetro	**Unidades**[3]	**Descripción**	**Ejemplo**
1	***Frecuencia Mínima***	lat/min	Frecuencia cardíaca (**FC**) mínima desde el paciente; por encima puede sincronizarse automáticamente el ***SAVI***.	040
2	***Frecuencia Máxima***	lat/min	**FC** máxima desde el paciente; por debajo puede sincronizarse automáticamente el ***SAVI***.	100
3	***Frecuencia Nominal***	bpm	Para cuando no haya sincronismo. En caso que La **FC** del paciente se encuentre por debajo de la frecuencia mínima, ó por encima de la máxima, el ***SAVI*** retoma el control y fija su tasa de bombeo e esta frecuencia nominal.	055
4	***Frecuencia Cambio Sincronismo***	lat/min	Cuando la frecuencia de bombeo del ***SAVI*** sincronizado se encuentra entre 40 y 80 bpm, el sincronismo es **1:1**, esto es, por cada latido del corazón nativo corresponde un bombeo del ***SAVI***. Si la frecuencia de bombeo del ***SAVI*** sincronizado se encuentra entre 80 y 100 bpm, el sincronismo es **2:1**, esto es, por cada dos latidos del corazón nativo corresponde un bombeo del ***SAVI***.	080
5	***Retardo de Contrapulsación***	en porcentaje (%) del ciclo cardíaco	Porcentaje desde el comienzo del ciclo cardíaco del paciente, sensado por el ***SAVI*** como aparición de una onda R del **ECG**, en que el sistema genera la sístole de la **BS**.	50
6	***Período Sistólico (%S)***	en porcentaje (%) del ciclo de bombeo	Porcentaje del ciclo de bombeo de la **BS** que dura la expulsión sistólica, independientemente si el ***SAVI*** está en funcionamiento sincrónico o asincrónico.	30

En Modo Sincronizado, si la frecuencia del corazón del paciente está entre 40 y 100 lat/min (rango de sincronización), por ej. 60 lat/min, el SAVI estará sincronizado con la onda R bombeando sangre a una frecuencia de 60 bpm y contrapulsando a los 500 ms a partir del pico R, con un período sistólico de 300 ms. En el Modo Sincronizado, y dentro del rango de sincronización, el retardo de contrapulsación y el período sistólico son proporcionales al período cardíaco y se expresan en porcentaje de dicho período. Así, la frecuencia de bombeo y la duración del movimiento sistólico de la BS variará momento a momento según la FC registrada desde el paciente**.** En Modo Sincrónico, si la frecuencia cardíaca del paciente está por fuera del rango de sincronización, esto es, por encima de la frecuencia máxima o por debajo de la frecuencia mínima ingresadas al microprocesador, o el corazón se vuelve arrítmico, el MCP toma el control y ordenará al SN del MP que trabaje automáticamente con una frecuencia de bombeo a la frecuencia nominal

[3] *lat/min = latidos por minuto, esto es la frecuencia cardíaca del corazón del paciente.*
bpm = bombeos por minuto, esto es accionamiento del SAVI por minuto.

anteriormente programada, en este caso, 55 bpm. Bajo estas condiciones el SAVI actuará en forma asincrónica con el ECG del paciente. En consecuencia se perderá el poder de contrapulsación del SAVI y el período sistólico corresponderá a la frecuencia nominal establecida, según el valor de %S ingresado. Para la Sincronización de la ACCM con el ECG del paciente se cuenta con un electrodo epicárdico (sobre la superficie externa del corazón) se implanta en el ventrículo izquierdo y se une con un conductor que recorre paralelamente y asilado el tramo electroneumático del SAVI. A su vez, a dicho tramo se ha incorporado otro conductor que hace a su vez de electrodo de referencia. Ambos electrodos, a través de una ficha a la salida del cuerpo del paciente, se conectan por un cable unipolar enmallado a la UEN. La presencia de dos entradas para el ECG se debe a que se puede seleccionar la polaridad de la onda R. Este sistema de *ACCM* al tener incorporado un sistema de sincronización con el ECG del paciente, permite que el ventrículo artificial pueda eyectar su contenido de sangre durante el periodo diastólico y de esta manera, actuar como un *contrapulsador*.

El sistema además cuenta con las siguientes capacidades:

3.1 Comando automático: En caso de detectar un funcionamiento irregular del MCP, se puede derivar el control del driver al Control Secundario (CS) del MP. Esta operación se realiza oprimiendo el pulsador Automatic Mode, haciendo que la UEN funcione en Modo Automático, en forma asincrónica. Los circuitos electrónicos encargados de controlar a cada Grupo del Sistema Neumático SN, son los siguientes: (1) Timer que conmuta en forma alternada por períodos de 15 min entre un Grupo y el otro, (2) Control Secundario de las válvulas electro-neumáticas, (3) Sistema de Control de Presión Neumática de los compresores. El sistema posee internamente dos microventiladores (cooler) de funcionamiento continuo, para mantener ventilado el interior, y el ingreso de aire al gabinete se encuentra filtrado. El conjunto neumático de cada Grupo del SN, tiene su propia electrónica de control y su fuente de alimentación, por lo que el funcionamiento de cada uno de ellos es totalmente independiente del otro.

3.2 Selectror de Grupo Neumático: posee una llave de tres posiciones: *Grupo 1 continuo* (1), *Alternante* (Auto) y *Grupo 2 continuo* (2). Estando la llave en la posición central (alternante), el timer funciona como se describió anteriormente, conmutando entre ambos Grupos. Ante la posible falla de un grupo, por ej., el Grupo 1, si bien el pasaje es automático, se puede pasar también manualmente a la posición Grupo 2 continuo, anulando la función de conmutación del timer quedando este grupo en funcionamiento permanente hasta que acuda el personal de reparación.

3.3 Control Secundario (CS): es un circuito que activando el pulsador Automatic Mode del panel frontal, la UEN pasa al Modo Automático (Fijo Secundario) (simultáneamente se enciende un indicador luminoso -rojo- ON). En este Modo se conmuta el control de la válvula electro-neumática del Grupo que se encuentra funcionando al CS[4], y el funcionamiento es fijo y totalmente independiente del ECG del paciente, es decir, siempre será asincrónico. Mientras el control de la válvula provenga del MCP y el pulsador Automatic Mode no se encuentra oprimido, el CS permanece inhabilitado. En caso que el *circuito de monitoreo de la batería* del MCP detecta que la batería que alimenta a este módulo se encuentre baja, este circuito deriva automáticamente el control de la UEN al Circuito Secundario.

3.4 Sistema de Control de Presión Neumática: El aire comprimido que imprime presión en las cámaras neumáticas, proviene en cada Grupo desde un motocompresor (12 V_{CC}, 30W). Está constituido por un motor de corriente continua que mueve un compresor a membrana, tomando aire por un conducto conectado a un tanque acrílico cilíndrico de vacío y liberándolo a otro tanque igual de presión. Como se señaló anteriormente, los tanques son compartidos por ambos Grupos. *Control de la Presión Neumática positiva*. Mediante un Control de Nivel de Presion Neumatica que se

[4] *Realmente el control se efectúa simultáneamente a ambas válvulas, pero el timer habilita a una sola de ellas.*

encuentra en el panel frontal de la UEN, se puede regular en cada Grupo neumático la presión necesaria para actuar sobre la prótesis cardíaca; desde 0 mmHg hasta 240 mmHg.

3.5 Control de la Presión Neumática negativa: El nivel de vacío se regula mediante un control electrónico que dirige el funcionamiento de una válvula reguladora de caudal **nivel de vacío**, que permite tomar aire fresco, a través de un filtro de aire para evitar el ingreso de partículas al sistema. El control de las Presión Neumática es un sistema realimentado que consiste en sensar la presión de aire en el tanque de presión mediante un transductor conectado al circuito electrónico, de manera que al aumentar la presión en dicho tanque, el circuito tiende a bajar las revoluciones del motor, disminuyendo el caudal de aire. Por el contrario si la presión es baja, el circuito aumenta las revoluciones, con el fin de mantener la presión sistólica constante alrededor de un valor elegido con el potenciómetro nivel de presión neumática.

Se muestra en la Figura 2, una vista general de la Unidad Electroneumática Ambulatoria (sin cubierta externa) y el detalle del panel frontal. La parte superior consta del Módulo de Control Principal y del Sistema Neumático duplicado. El componente inferior consta del cargador de baterías. El conjunto está montado sobre un carrito con ruedas para facilitar su desplazamiento.

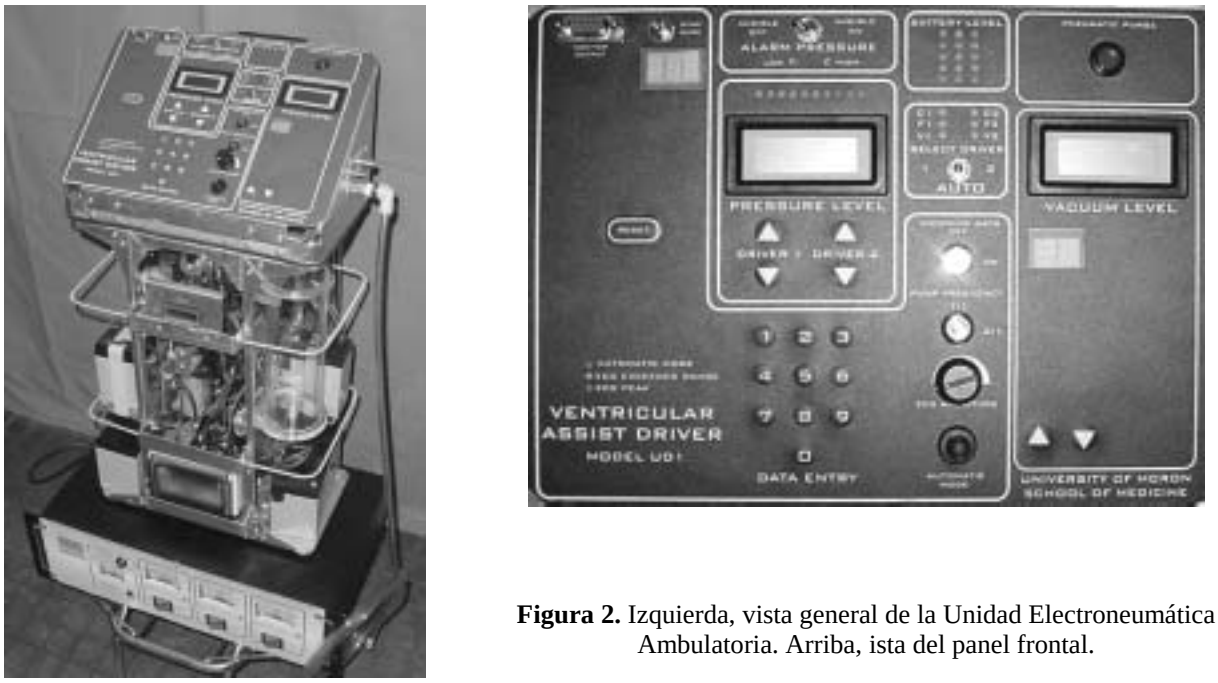


Figura 2. Izquierda, vista general de la Unidad Electroneumática Ambulatoria. Arriba, ista del panel frontal.

4. MONITOREO REMOTO

Para tomar y convertir los datos fisiológicos del paciente y los del funcionamiento del driver, se desarrollo e implemento un Sistema de Adquisición de Datos (SAD) por comunicación serie, para posterior monitoreo en forma remota, recepción y registro en bases de datos de la información enviada por el SAD a través de una computadora. El sistema se conectara vía modem de alta velocidad tipo ADSL o cable modem, a Internet e Intranet. Se emplea SQL Server, Microsoft Personal Web, Server e Internet Informatión Server para la explotación y administración de información de pacientes. El sistema permite visualizar la información registrada por el SAD. El desarrollo se enfoca a lenguajes de alto nivel (Java, .Net, C++), que permiten la construcciones de aplicaciones en tres capas.

Se comenzó investigando como monitorear mediante una computadora configurada como "visor o controlador" a otra computadora configurada como remota, es decir, a ser monitoreada o controlada independientemente de la ubicación geográfica de ambas máquinas. En un principio se utilizó el programa Real VNC, el cual fue ensayado entre dos máquinas próximas y distantes, y se comprobó que posee unas prestaciones adecuadas para los requisitos que inicialmente se desean cumplir. Este es un programa gratuito mediante el cual se puede controlar completamente con un mouse y teclado de una computadora o PC configurada como Viewer a otra computadora remota en el que previamente se encuentra instalada esta aplicación y que está configurada como Server. Este paquete instala el servidor (Server) que es necesario en la computadora que va a ser controlada o monitoreada y el cliente (Viewer) que será el que tendrá que ejecutarse en la computadora desde donde se va a ejercer el control.

El funcionamiento es muy sencillo: se ejecuta el Viewer, se introduce la IP de la computadora que se quiere controlar o monitorear remotamente, y se introduce la contraseña, a continuación se tiene en una ventana el escritorio de la otra computadora con el control de su mouse y escritorio. Se podrá establecer contraseña de acceso en el servidor, visualizar la computadora controlado en ventana o pantalla completa, utilizar diferentes resoluciones e incluso habilitar el control compartido desde varios clientes. Siendo esta opción, para el inicio de este proyecto de investigación la solución que permite a un médico y una computadora utilizada como cliente monitorear un paciente a través de una computadora remota como servidor estando éste cualquier punto geográfico próximo o distante al médico que lo chequea. Una vez adquirida la experiencia, se realizó un ensayo con un SAVI funcionando en condiciones in vitro en el laboratorio. Se desarrolló un programa que simula el monitoreo del ECG de un paciente. Así, en este ensayo, los parámetros graficados a medir son: la forma de onda del ECG y la presión neumática generada por la UEN. Las señales son leídas por un programa en Visual Basic que lo reproduce gráficamente y en tiempo real como si a la paciente fuera sido monitoreado en el momento. También esos archivos pueden ser visualizados y graficados mediante Excel. Cada vez que la lectura de los archivos llega al último valor, se reinicia para simular un proceso continuó. La Figura 3 muestra la representación del sistema de la lectura de valores máximos y mínimos de presión neumática y frecuencia cardíaca. Se construyó además un circuito que adquiere información de ECG y presión neumática de la UEN y que lo transmite por RS-232 a la computadora, para que esa información sea graficada y simultáneamente monitoreada en una estación remota vía Intranet/Internet. También informará acerca de los parámetros programados en la UEN (Presión Máxima, Mínima, Nominal, Porcentaje Sístole, porcentaje Contrapulsación). La Figura 4 muestra en forma integrada la lectura de valores máximos y mínimos de presión neumática y frecuencia cardíaca y el estado del dirver, como así también los valores ingresados a cada parámetro del sistema.

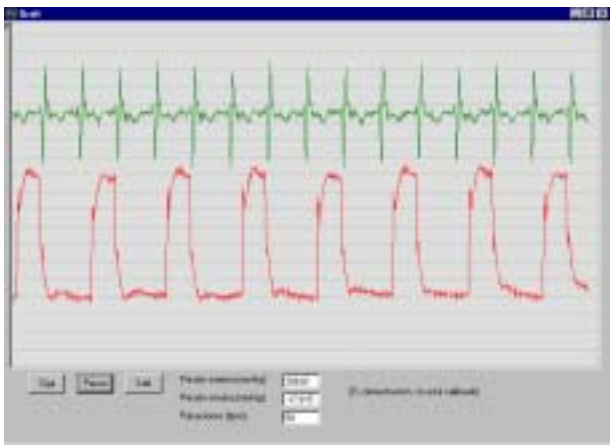


Figura 3. Lectura de valores máximos y mínimos.

Figura 4. Representación integrada valores de presión neumática y frecuencia cardiaca, estado del dirver y parámetro ingresados.

5. ARQUITECTURA BASE DEL SISTEMA

Las arquitecturas para aplicaciones en el marco del monitoreo remoto de pacientes requieren de flexibilidad esto hace que la aplicaciones sean extensible en funcionalidades, en el ámbito de la telemedicina el propósito de un Framework, es brindar una alternativa más dinámica, respecto al monitoreo de los sistemas ECG tradicionales, e incentivar a los profesionales de la salud a incorporar el uso de las nuevas tecnologías. A continuación se expone las interrelaciones con el hardware y el software definidas en el framework. En la Figura 5 se observa la distribución de los distintos módulos de software con respecto a los componentes que integran la arquitectura de hardware [2]. Todos aquellos dispositivos englobados bajo el nombre de "Periféricos de Adquisición de Datos" (PAD), tienen como tarea la obtención de señales de datos analógicas desde los pacientes, y luego la transformación a señales digitales para poder interactuar con las "Terminales de Adquisición de Datos" (TAD). Las TAD, cumplen el propósito de transformar la información digital, proveniente de los PAD, a registros de datos para guardarlos dentro del Servidor de Base de Datos. El Servidor Web contiene aquellas Aplicaciones que brinda el Framework, ya sean de Administración, Monitoreo, y Web Services. Mediante los Web Services, se permite la adquisición de datos almacenada en el servidor de Base de Datos, y la interacción con diferentes plataformas que deseen comunicarse con nuestro Framework. Las "Terminales Monitoras" (TM), tienen como propósito poder exhibir la información que se encuentra dentro de la base de datos, acorde a los parámetros indicados por el usuario previamente. En cuanto a las tareas que realizan cada uno de los módulos de Software, se detallan en la siguiente sección. El desarrollo de la arquitectura de monitoreo en este caso para ECG se basa en la utilización de tecnología Microsoft para todos los elementos que componen al sistema. El sistema operativo de base para los de servidores de Adquisición, Servidores Web es Windows Server 2003 debido a su alto grado de confiabilidad y estabilidad Como motor de Base de Datos utilizaremos SQL Server 2005

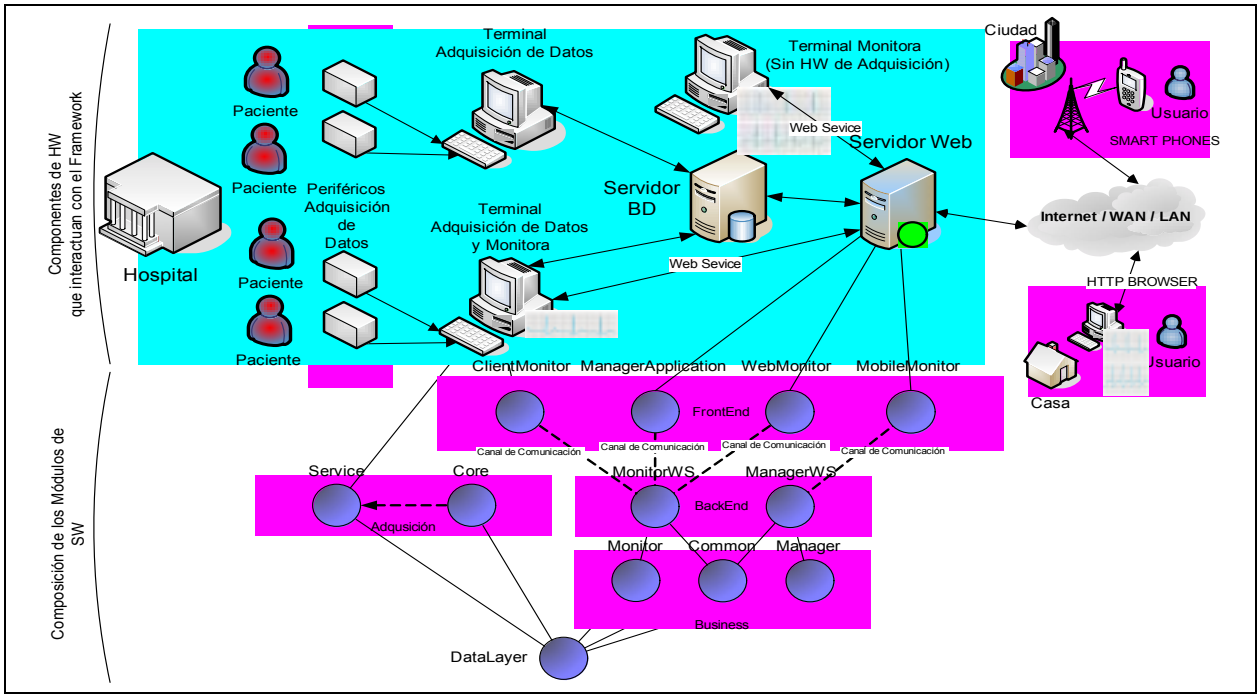


Figura 5: Alcance esquemático del framework.

El equipamiento y distribución ideal es el detallado debajo Figura 6 en el que se expone claramente el escenario en el cual se ejecutan todos los módulos del sistema.

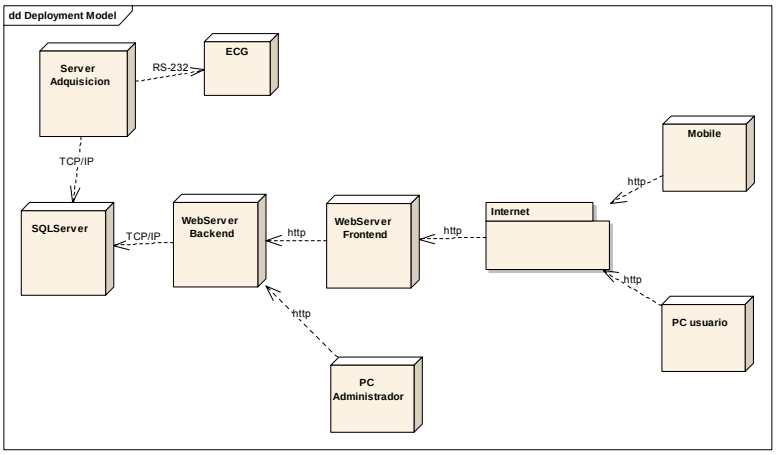


Figura 6: Diagrama de Despliegue

6. CONCLUSIÓN Y FUTURAS LÍNEAS DE INVESTIGACIÓN

En el contexto de la telemedicina, el sistema de Sistema de Asistencia Ventricular Izquierda como el sistema de monitoreo y la arquitectura propuesta contribuyen a los sistemas médicos puede lograr una mayor eficacia de los servicios de salud y un mejor acceso a la atención de personas al mejorar la calidad de la atención médica y favorecer la salud, también ésta puede ser beneficiosa para los prestadores, los profesionales y los usuarios finales de la atención. Asimismo, debería influir en los costos de la atención, al reducir el número de exámenes médicos superfluos o repetidos. Más aún, en la mayoría de los países existen poblaciones alejadas y de difícil acceso y múltiples inconvenientes en la accesibilidad a la atención en salud.

Con los últimos desarrollos en nuestro sistema *Novel III LVAS*, es posible realizar un monitoreo *on line* del funcionamiento del driver y de los parámetros clínicos del paciente, desde un lugar remoto. Esto es posible a través de una conexión driver-ordenador mediante el uso de Internet/Intranet, tanto desde una sala de cuidados intensivos como desde el domicilio del paciente. En estos momentos el sistema está diseñado para el monitoreo clínico y fisiológico a distancia de parámetros corporales y estudios del paciente: ECG, PA, VM y temperatura corporal y el monitoreo electrónico remoto de diversos parámetros de funcionamiento del driver tales como valores de los parámetros programados, presión neumática, frecuencia de bombeo, temperatura de los compresores, nivel de sincronización, alarmas, estado baterías y grupo funcionando.

Como futuras líneas de investigación que surgen a raíz de este trabajo y que son potencialmente importantes se enumeran las siguientes: (1) Integrar mediante servicios web al Sistema con un Sistema Experto para la asistencia en Diagnósticos, (2) Aplicar Minería de Datos sobre la información recolectada para predicción de tendencias o comportamientos emergentes que permitan la toma de decisiones proactivas. (3) Extensión de funcionalidades al Sistema para monitorear más funciones vitales (4)

También se prevé explorar la infraestructura para la incorporación de consultas datos de la Historia Clínica y estudios ecocardiográficos, radiografías, etc., como así también la posibilidad de teleconferencia con el propio paciente.

7. REFERENCIAS

[1] Alberto Kopec, José Salazar Gómez. Aplicaciones de Telecomunicaciones en Salud en la subregión andina. *Oras-Conhu Organismo Andino de Salud, abril 2003. http://www. comunidadandina.org/telec/Documentos/Telecomunicaciones_salud.pdf*

[2] Andrada David, Sparhakl Pablo, Novillo Hernán, Ierache Jorge, Arquitectura para el Monitoreo Remoto de Funciones Vitales en Pacientes Ambulatorios, XII Congreso Argentino de Ciencias de la Computación octubre de 2006 - Potrero de los Funes, San Luis, Argentina Universidad Nacional de San Luis

[3] Carmen C. Y. Poon, Yuan-Ting Zhang. A Novel Biometrics Method to Secure Wireless Body Area Sensor Networks for Telemedicine and M-Health. IEEE Comunications Magazine, abril 2006.

[4] Cervino, C; Nasini, V; Sroka, A; Diluch, A; Cáceres, M; Sellanes, M; Malusardi, A; del Río, M; Pham, S and D Liotta. Novel left ventricular assist system I and II for cardiac recovery therapy: the driver. Tex Heart Inst J 2003; In Press.

[5] iPath Project - iPath association en colaboración con la Universidad de Basel, Suiza. http:// telemed.ipath.ch/ipath/

[6] Lin, James C. Aplicando tecnología de telecomunicaciones al cuidado de la salud: el estado actual y los cambios en la telemedicina. Engineering in medicine and biology. IEEE, Vol 18 N 4 Jul/Ag 1999, pp 28 –31

[7] Liotta D, Hall CW, Henly WS, Cooley DA, Crawford ES, DeBakey ME. Prolonged assisted circulation during and after cardiac or aortic surgery. Prolonged partial left ventricular bypass by means of intracorporeal circulation. Am J Cardiol 1963;12:399-405.

[8] Liotta D. Novel left ventricular assist system II. Tex Heart Inst J 2004;31:278-82.

[9] Liotta D. Novel left ventricular assist system. An electrocardiogram-synchronized LVAS that avoids cardiac cannulation. Tex Heart Inst J 2003;30:194-201.

[10] Luraschi, R. Sistema de Monitorización a Distancia de Señales Biológicas Vitales a través de Radio. 3er. Congreso Argentino de Informática y Salud – SADIO

[11] Magnus Berggren. Wireless communication in telemedicine using Bluetooth and IEEE 802.11b. OpenECG - Standard communications protocol for computer assisted electrocardiography. http://www.openecg.net/index_ws2.html

[12] OpenECG - Standard communications protocol for computer assisted electrocardiography. *http://www.openecg.net/index_ws2.html*

[13]PhysioNet – Public service of the Research Resource for Complex Physiologic Signals. Massachusetts Institute of Technology (MIT). http://physionet.org/.Uppsala University, Suecia, noviembre 2001. http://www.it.uu.se/research/publications/ reports/2001-028/2001-028.pdf.

[14] World Health Organization. Technical report article A58/21 1998. http://www.who.int /gb/ebwha /pdf_files/WHA58/A58_*21-sp.pdf*

Un algoritmo para la detección de horizonte en dispositivos voladores radiocontrolados

Pablo Odorico[1] y Claudio Delrieux[2]

[1] Departamento de Cs. e Ing. de la Computación
[2] Departamento de Ing. Eléctrica y Computadoras
Universidad Nacional del Sur - claudio@acm.org
Parcialmente financiado por la SECyT-UNS

Abstract

Presentamos un algoritmo para la detección no supervisada en tiempo real del horizonte en imágenes de video. El mismo está diseñado para ser implementado en dispositivos autónomos de bajo costo y capacidad de cómputo (microcontroladores), como para ser incorporados en vehículos voladores no tripulados (UAVs). Esto permitirá la obtención de información indispensable para la orientación, navegación, georeferenciamiento y la elaboración de mosaicos de imágenes.

El algoritmo se basa en una función de costo para los pixels de un submuestreo de la imagen en las coordenadas YI del espacio cromático YIQ. Con una baja cantidad de evaluaciones el algoritmo determina la orientación más probable del horizonte, y luego refina la búsqueda con un método adaptativo. La implementación de un prototipo en una PC muestra que el algoritmo es lo suficientemente robusto como para determinar la posición del horizonte con muy bajo costo computacional.

Palabras Clave: Procesamiento de Imágenes — Procesamiento de Video — Sensado Remoto

1. Introducción

La obtención de imágenes para sensado remoto (aéreas, satelitales, etc.) es siempre un recurso costoso y de disponibilidad limitada. Sin embargo, en zonas geográficas de accesibilidad limitada, estas imágenes representan el único medio primario de obtención de información. Las imágenes satelitales son caras, están disponibles para fechas y condiciones climáticas no negociables. La resolución es aceptable, aunque imágenes satelitales de resolución comparable con las obtenibles con imágenes aéreas (p. ej., imágenes IKONOS) son de un costo prohibitivo.

Las imágenes aéreas, por su parte, se pueden generar para fechas y condiciones climáticas flexibles, pero el costo operativo es muy alto, requieren un post-procesamiento intensivo para georeferenciarlas, así como corregir la distorsión por perspectiva, y ecualizar las luminancias.

Una solución para mejorar el costo operativo consiste en utilizar dispositivos voladores teledirigidos, montando en ellos una cámara con un microcontrolador para realizar la adquisición. El georeferenciamiento se puede conseguir agregando un GPS a la electrónica anterior, lo cual es posible actualmente dado el bajo costo y la alta performance de estos dispositivos. La corrección de la distorsión por perspectiva, por su parte, requiere determinar en tiempo real la orientación de la cámara. Esto se determina por medio de la localización del horizonte.

Los vehículos voladores no tripulados (UAVs) se caracterizan por su bajo momento de inercia, por lo cual son susceptibles a rápidos cambios de orientación y velocidad. Por lo tanto, para garantizar la estabilidad del vuelo, y así mismo del *tracking* de la toma de video, la determinación de la posición del horizonte es esencial [6]. Sin embargo, este es un proceso que puede resultar computacionalmente costoso, teniendo en cuenta las limitadas capacidades del hardware a ser incorporado en el dispositivo volador.

En este trabajo proponemos un algoritmo rápido y preciso para la determinación no supervisada del horizonte. El mismo se basa en evaluar un subconjunto submuestreado de la imagen original. Para cada una de las muestras o pixels examinados, se determina la posición de los mismos en un subespacio cromático YI, evaluando una función de similitud (determinada experimentalmente) con las coordenadas del cielo en dicho espacio. Un algoritmo de tracking recorre las muestras en cuatro posibles direcciones, determinando la orientación óptima de una frontera dinámica entre pixels clasificados como probable cielo y pixels clasificados como improbable cielo. Por último, un ajuste lineal de la frontera encontrada produce el horizonte detectado.

El algoritmo fue implementado en PC, y testeado con videos de prueba tomados con una cámara en movimiento sobre un lugar elevado, simulando condiciones similares a las que se encontrará en el dispositivo volador. Los resultados obtenidos son satisfactrios, y el tiempo de cómputo estimado está completamente dentro de los límites de un microcontrolador moderno, por lo que se estima que la implementación de un prototipo permitirá reproducir estos resultados. Se presentan también mejoras e ideas que permitirán reducir aún más el tiempo de cómputo así como aumentar la precisión de los resultados.

2. Trabajo Previo

La navegación teleasistida de vehículos voladores no tripulados (UAVs) es un tema de creciente interés. En [3, 4] se presenta un algoritmo de estabilización y navegación autónoma guiado por visión computacional, el cual se basa en un método robusto para determinar la posición del horizonte. Este método de detección del horizonte se fundamenta en resolver el siguiente problema de optimización. Sea una recta $y = ax + b$ en coordenadas del viewport, se superpone esa recta a la imagen y se clasifican las muestras del color en el espacio cromático RGB de los

pixels por encima o por debajo de la recta. Para cada grupo de muestras, se computa la varianza 3D en las tres coordenadas del espacio cromático. El método, entonces, busca los valores de pendiente a y ordenada al origen b que minimiza la suma de las varianzas de ambos grupos de muestras.

Si bien la hipótesis de partida de este método es valiosa (el horizonte es una línea que separa dos zonas homogéneas en un frame o cuadro del video), es también evidente que es computacionalmente costoso. Contando con un enlace de radio es posible transmitir el *stream* de video y operar la navegación autónoma con una computadora en tierra. Sin embargo, nuestro propósito consiste en resolver la navegación con un microcontrolador a bordo, por lo cual la complejidad del este método lo hace inadecuado.

En el otro extremo del espectro, es posible pensar en chips dedicados a la detección del horizonte, como la propuesta presentada en [2]. Este integrado provee un sensor de 12×12 fotorreceptores, e implementa un algoritmo de separación lineal basado en la luminancia de los receptores. Si bien los tiempos y los consumos de potencia del integrado son sobresalientes, el algoritmo es rígido respecto de los parámetros dentro del cual opera. Por ejemplo, sería imposible realizar vuelos nocturnos.

Nuestra propuesta, en cambio, busca un equilibrio entre ambas posibilidades (computación en un chip vs. computación en tierra), utilizando dispositivos programables más flexibles y algoritmos de bajo costo. Eventualmente, en caso de ser necesario, los algoritmos pueden ser migrados a dispositivos más eficientes como los FPGA.

3. Descripción del método

Como ya mencionáramos, el método consiste en trabajar con un subsampleo de la imagen, de tamaño determinado dinámicamente de acuerdo con las restricciones que encuentre el algoritmo en tiempo de ejecución. Una mayor cantidad de muestras permite una determinación más precisa del horizonte, con un costo mayor, aunque es más sensato trabajar con una cantidad baja de pixels y luego refinar adaptativamente en las áreas donde se requiere mayor precisión. Para los ejemplos desarrollados en este trabajo, las imágenes originales de 640×480 se submuestrearon a $\downarrow 32$ por coordenada, resultando una resolución de 20×15 para cada cuadro del submuestreo , lo cual resultó ser lo más adecuado en los casos que fueron analizados

Los pixels del submuestreo (o muestras) se pre-clasifican utilizando como función de costo a una "distancia" d en espacio cromático YIQ [1],

$$d(I, Y) = (2I - Y + 1{,}2)^2,$$

la cual resulta de evaluar la distancia Euclídea al cuadrado entre la posición del color del pixel y la recta $2I - Y + 1{,}2$ en el espacio YIQ[1]. Una distancia menor que 0.2 determina que el contenido del pixel es de un color típicamente cercano al cielo. Es importante destacar que como

[1]El valor lineal de la distancia hubiera sido igualmente eficiente para determinar una clasificación, pero el uso de la distancia al cuadrado permite ahorrar una operación de punto flotante por cada muestra considerada.

$$d(p_1) < u$$

$$d(p_2) \geq u$$

$$p_a = \frac{u - d(p_1)}{d(p_2) - d(p_1)} p_1 + \frac{d(p_2) - u}{d(p_2) - d(p_1)} p_2$$

Figura 1: Determinación de un punto de frontera entre dos muestras.

el algoritmo es incremental, no se requiere convertir todos los pixels del cuadro submuestreado a YIQ y calcular el costo, sino solo a un subconjunto dinámicamente determinado de ellos, lo cual reduce el costo computacional sensiblemente. La función de costo, y el umbral mencionado más arriba fueron determinados experimentalmente como los valores que permiten una mejor separación entre las muestras que pertenecen al cielo y los que no.

En una primer etapa, se realiza una estimación de la orientación más probable del horizonte, dividida en cuatro cuadrantes que van de -45 a 45 grados respecto de la horizontal, de 45 a 135, de 135 a -135, y de -135 a -45. Para determinar el cuadrante más probable se hace el cálculo para los cuatro casos, y se elige el caso en el cual el horizonte queda mejor definido. Cada orientación gruesa determina un recorrido de las muestras, respectivamente descendente por columnas, por filas de izquierda a derecha, ascendente por columnas, y por filas de derecha a izquierda.

Por ejemplo, para testear si la orientación más probable del horizonte está entre -45 y 45 grados respecto de la horizontal, se recorren las muestras por columna en sentido descendente, asumiendo que el valor de d de las muestras inferiores están por encima del umbral (de no ser así, se descarta la columna por "anómala"). El recorrido en cada columna termina cuando se cruza el umbral 0.2 en el d, lo cual se considera que ocurre cuando un pixel cruza el horizonte que buscamos. La diferencia entre los d de los dos pixels sucesivos en la columna donde se produjo la transición da una idea de la "confiabilidad"de ese sitio como un punto probable del horizonte. Para cada una de las cuatro posibles orientaciones, la "confiabilidad global"es el acumulado de la confiabilidad de cada uno de los puntos de transición. Finalmente, la orientación más probable es la que obtuvo mejor confiabilidad global.

Los puntos de frontera se computan por interpolación lineal, utlizando la misma idea subyacente del algoritmo "marching squares"[7] (ver Fig. 1). Es importante notar que en un primer momento intentamos realizar el tracking del horizonte por medio de este algoritmo, obteniendo resultados muy pobres y problemáticos, lo cual derivó en la idea que finalmente presentamos aquí. También cabe resaltar que nuestro algoritmo procesa localmente la parte de la imagen submuestreada que es atravesada por el horizonte. En caso de ser necesaria mayor exactitud, el algoritmo puede recurrir a la imagen en un subsampleo más preciso del cuadro original, obteniendo una poligonal de mejor resolución.

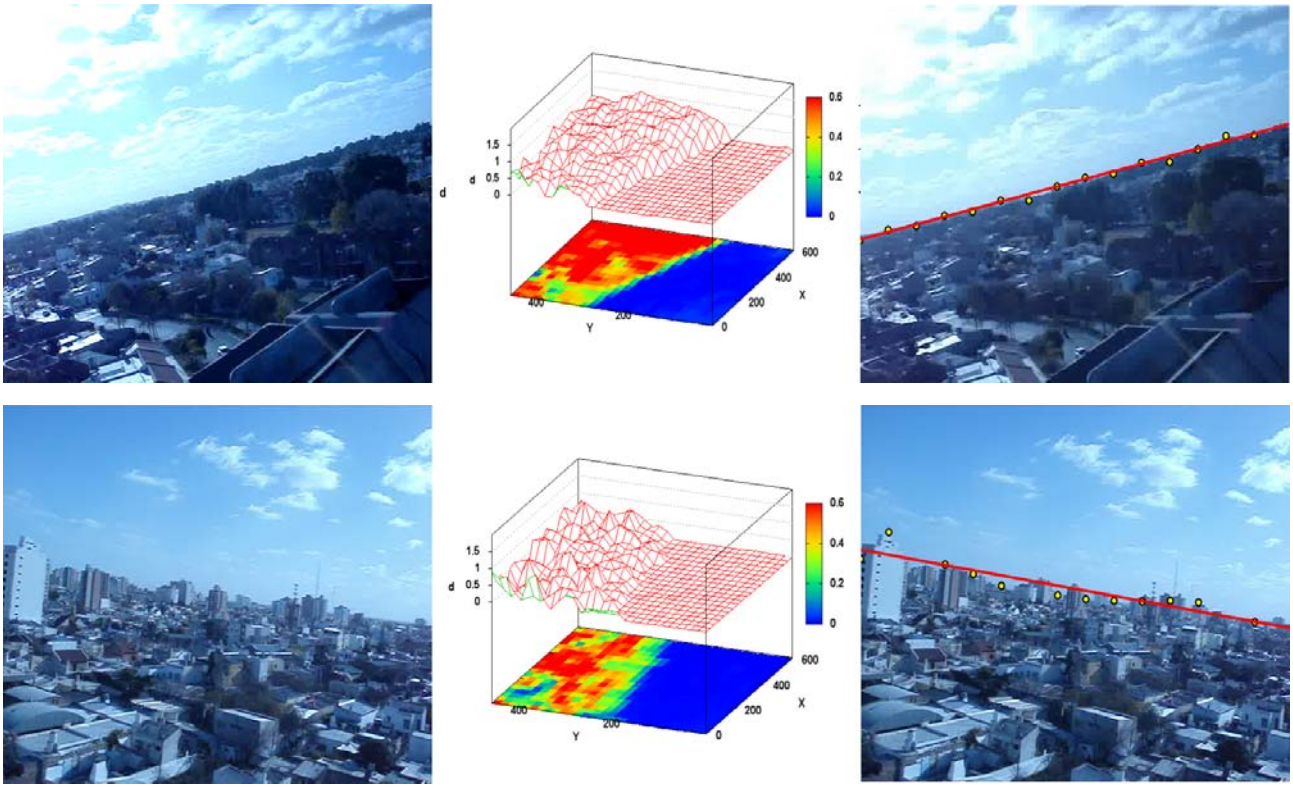


Figura 2: Ejemplos del horizonte encontrado con nuestro método.

En una segunda etapa, los puntos de frontera obtenidos pueden unirse por una poligonal, o bien puede computarse un ajuste lineal, lo cual determina un horizonte promedio que oblitera los obstáculos que pueden presentarse en la línea visual. También es posible deducir la inclinación de la cámara con respecto al plano teniendo en cuenta la elevación media de los puntos del horizonte. En la Fig. 2 es posible ver dos cuadros de un video, la función de costo en cada muestra de la imagen submuestreada, los puntos de frontera encontrados, y el ajuste de estos últimos por regresión lineal.

4. Resultados obtenidos

La performance del algoritmo resultó sobresaliente, tanto en tiempo de cómputo como en la calidad de los resultados. Implementamos un prototipo del método en PC, el `Horizon 1.0`, para poder evaluar los resultados y visualizar sus propiedades. En la Fig. 3 se muestran los resultados en un cuadro de prueba, obtenido de una toma de video con una cámara fotográfica comercial, sin corrección de la temperatura cromática [5], lo cual dificulta aún más la detección del horizonte en tomas diurnas.

En la interfaz del prototipo se muestra el cuadro original, al que se superponen los puntos de frontera obtenidos con el algoritmo. También se muestran la distribución del color de los pixels en el plano YI del espacio cromático YIQ, la superficie de costo obtenida interpolando la función d definida en la Sección anterior sobre todas las muestras en el cuadro, y la inclinación

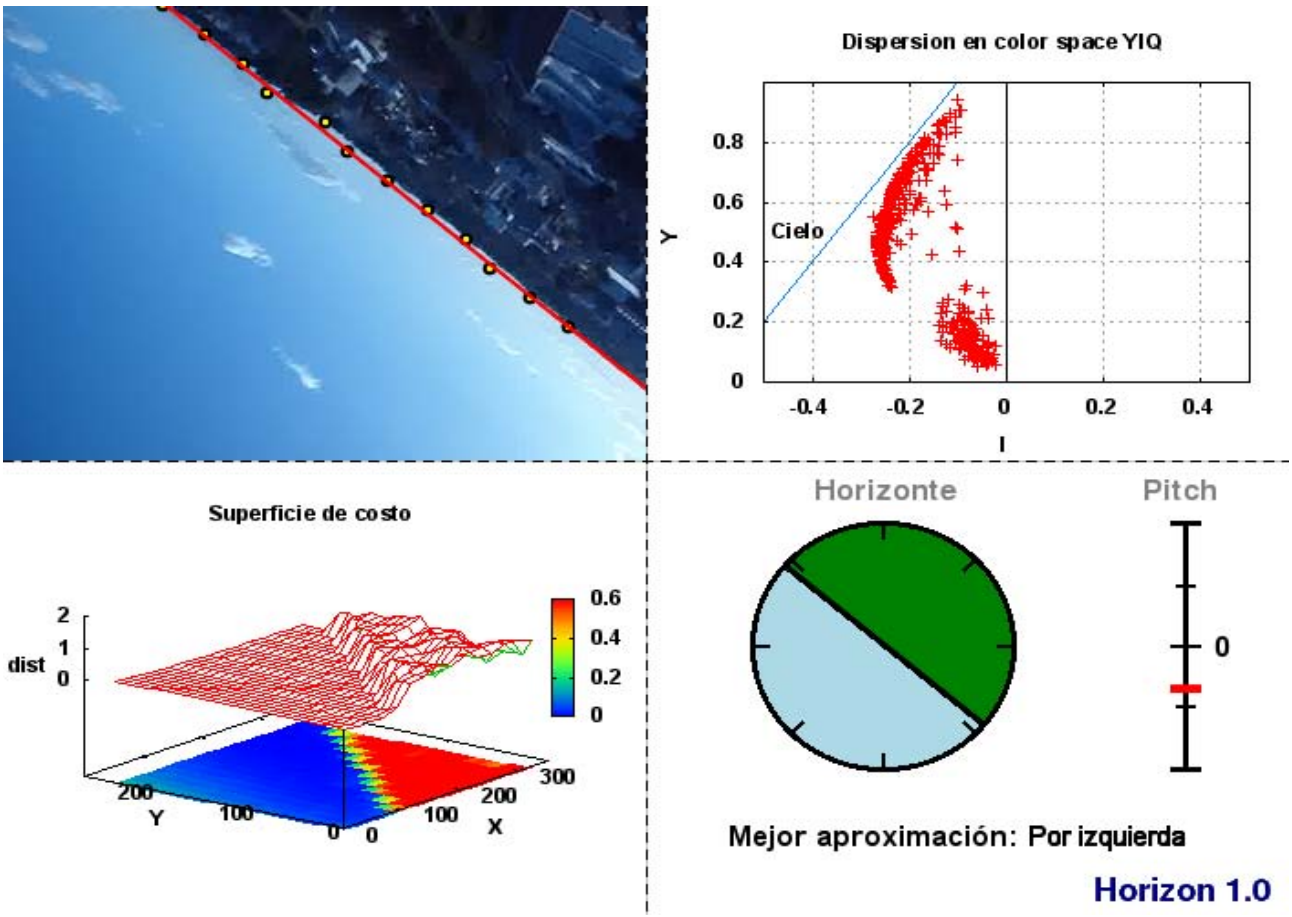


Figura 3: `Horizon 1.0` y un cuadro de prueba.

y posición del horizonte virtual. En la Fig. 4 se aprecia cómo el algoritmo de búsqueda de fronteras reacciona bien frente a los obstáculos, en este caso edificios.

El algoritmo está en condiciones de determinar cuándo no existe una línea de horizonte probable. En la Fig. 5 se tomó como cuadro de prueba una parte del video en la cual la cámara apunta hacia el cielo. La primer parte del algoritmo encuentra que no hay puntos que cruzan el valor umbral predeterminado en la función de costo, por lo que la salida del algoritmo es un horizonte nulo.

En la siguiente tabla se consignan las velocidades aproximadas de ejecución para los cuadros en los que la detección del horizonte resultó la más rápida o la más lenta. En los peores casos, con una PC antigua de bajo poder computacional, el método computa varios miles de cuadros por segundo, lo cual permite concluir que su implementación en microcontroladores modernos tendrá una performance más que suficiente para los objetivos requeridos.

CPU	Mejor caso	Peor caso
Pentium 2 MMX	25 Khz.	2 Khz.
Athlon64 3000+	1600 Khz.	16 Khz.

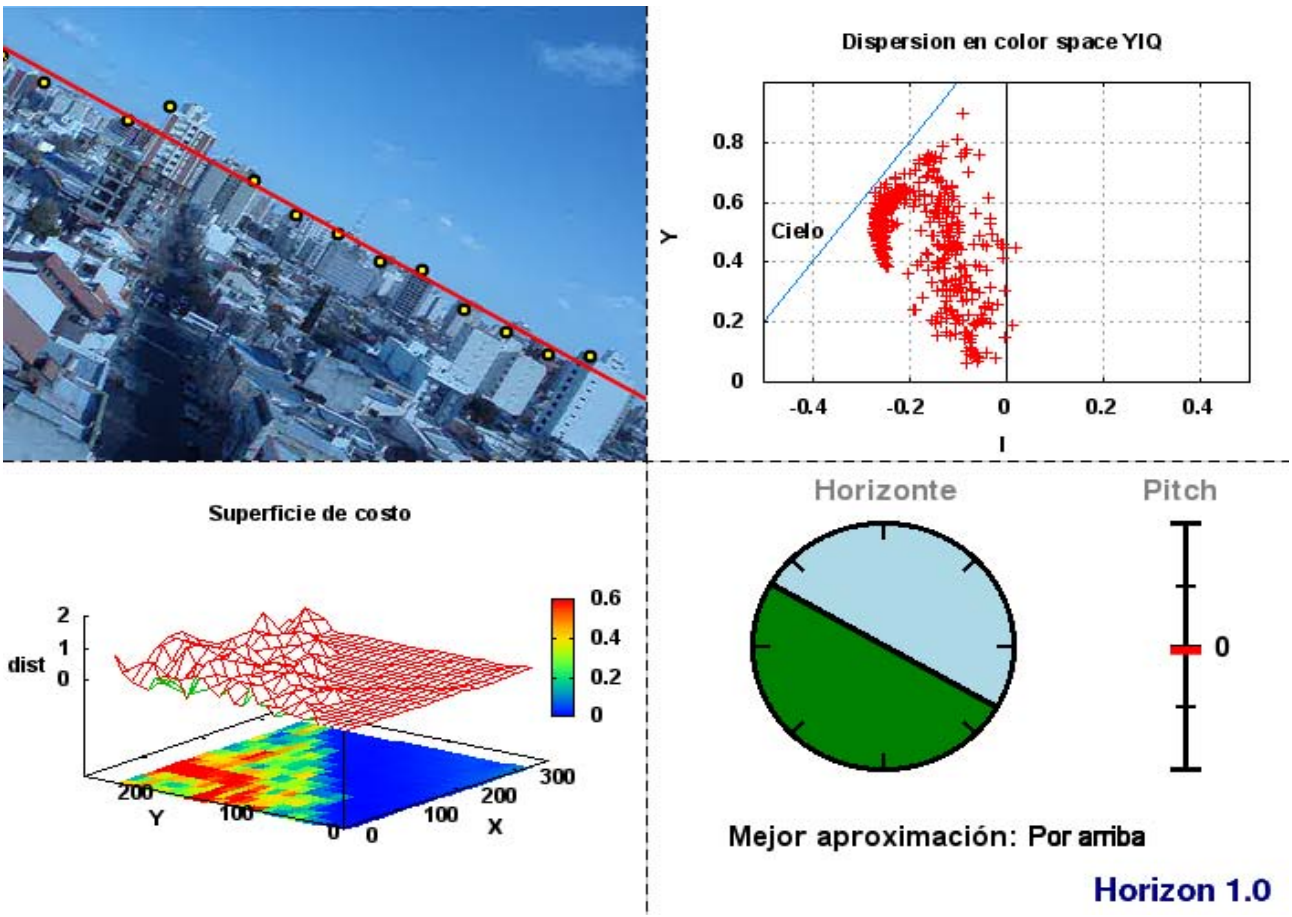


Figura 4: Horizonte con obstáculos.

En `http://www.lip.uns.edu.ar/horizonte` es posible ver un conjunto de videos completos con los resultados del método en tiempo real, tomados en diferentes condiciones y con distintos tipos de obstáculos.

5. Conclusiones y trabajo futuro

Se presentó un algoritmo robusto para la determinación en tiempo real del horizonte en imágenes de video. El mismo se basa en clasificar el color de los pixels por medio de una función de costo en el espacio cromático YIQ. En un primer paso, el método determina la orientación gruesa más probable del horizonte, por medio de una búsqueda del punto de cruce de un valor umbral predeterminado en la función de costo. Luego, el algoritmo realiza un *tracking* entre muestras sucesivas, buscando los puntos de frontera de acuerdo a la función de costo. Finalmente, los puntos de frontera del paso anterior son ajustados linealmente para determinar la posición del horizonte y su inclinación respecto del plano de tierra.

El algoritmo fue implementado en un prototipo en PC en una plataforma de programación de distribución libre, exhibiendo una performance muy buena, tanto la calidad del resultado como en tiempo de ejecución. Los resultados obtenidos en tomas de video muestran que el

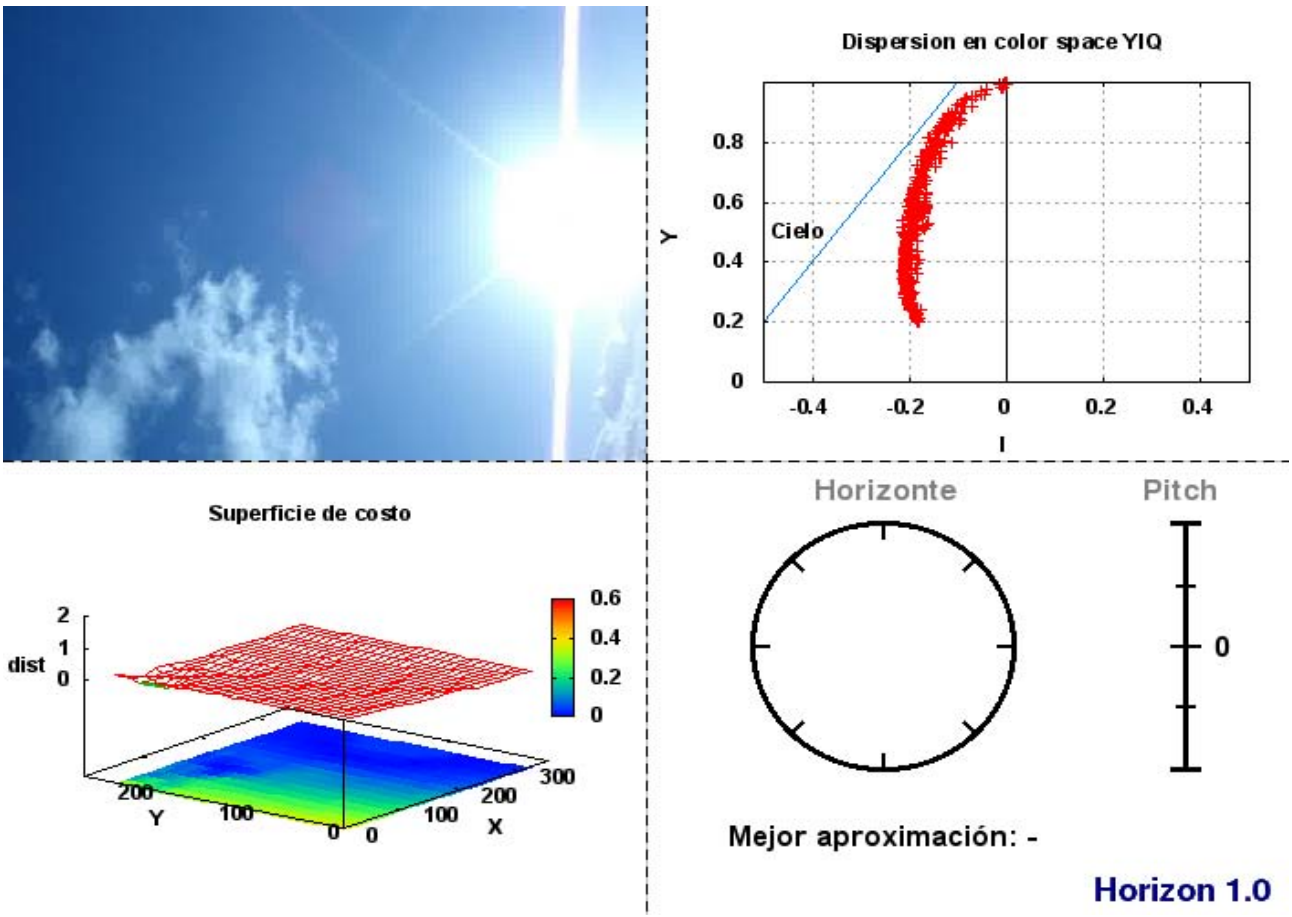


Figura 5: Cuadro de prueba sin horizonte.

algoritmo tiene la capacidad de determinar la posición del horizonte en tiempo real en forma robusta, así como determinar fronteras no lineales que rodean a obstáculos, o decidir si no existe línea de horizonte en un cuadro determinado.

Entre las ventajas determinantes de nuestro algoritmo, podemos mencionar que su implementación computacional es sencilla, por lo que la migración a dispositivos microcontroladores es inmediata. Además, es lo suficientemente flexible como para poder adaptarse a diferentes contextos de uso. En particular, claramente es posible determinar en forma dinámica una relación de compromiso entre la cantidad de cuadros por segundo procesados y la resolución o calidad de la frontera encontrada. Estas dos ventajas lo destacan respecto de los demás algoritmos publicados hasta ahora.

Como trabajos futuros a realizar, podemos contar la migración del algoritmo a un microcontrolador para determinar si las velocidades de cómputo son adecuadas, y su inclusión dentro de un sistema de navegación asistido. También es de gran importancia poder determinar si el algoritmo es factible de ser utilizado con imágenes infrarrojas, dado que en este caso sería necesario encontrar una función de costo adecuada. Por último, debería ser posible optimizar la búsqueda de la frontera tomando en cuenta la información del último cuadro procesado. Esto permitiría saltear el primer paso del algoritmo y de esa manera aprovechar la capacidad de

cómputo para obtener más cuadros por segundo, o bien mejorar la resolución de la frontera obtenida.

Referencias

[1] Rafael González and Richard Woods. *Digital Image Processing (Second Edition)*. Addison-Wesley, Wilmington, USA, 2002.

[2] T. Horiuchi. A Low-Power Visual Horizon Estimation Chip. In *Proceedings of the International Symposium on Circuits and Systems*, pages 4755–4758, Kobe, Japan, 2005. International Circuits and Systems Society, ISCAS.

[3] S. M. Ettinger. *Design and Implementation of Autonomous Vision-Guided Micro Air Vehicles*. PhD thesis, University of Florida, 2001.

[4] S. M. Ettinger, M. C. Nechyba, P. G. Ifju and M. Waszak. Vision-guided flight stability and control for micro air vehicles. *Advanced Robotics*, 17(1):617–640, 2003.

[5] Wikipedia. Color Temperature. `http://en.wikipedia.org/wiki/Color_temperature`, 20-7-2007.

[6] Wikipedia. Horizon. `http://en.wikipedia.org/wiki/Horizon`, 20-7-2007.

[7] Wikipedia. Marching Squares. `http://en.wikipedia.org/wiki/Marching_squares`, 20-7-2007.

Un modelo cognitivo para un agente BDI con compromiso social

Sonia V. Rueda
svr@cs.uns.edu.ar

Guillermo R. Simari
grs@cs.uns.edu.ar

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
8000 Bahía Blanca, Argentina

Abstract

Este trabajo propone un modelo para representar el conocimiento y el comportamiento de un agente BDI que actúa e interactúa condicionado por las normas y la estructura social de la organización a la que pertenece. Parte del conocimiento está compartido globalmente y algunos elementos cognitivos son específicos de cada individuo en particular. Cuando el conocimiento de un agente es insuficiente para alcanzar su meta comprometida inicia un diálogo colaborativo. La autonomía y reactividad le permiten adaptarse a las características dinámicas del entorno. La estructura normativa y la asignación de roles brindan patrones para anticipar la evolución de los diálogos.

Keywords: Sistemas Multiagente, Programación en Lógica, Argumentación, Diálogos Colaborativos, Roles, Normas

1 Introducción

Un gran número de aplicaciones desarrolladas siguiendo el modelo de agentes demandan altos niveles de colaboración. La colaboración ofrece un ambiente cooperativo que aumenta las posibilidades para que cada individuo alcance sus metas, pero también provoca interferencias y dependencias entre las tareas, exigiendo que las acciones sean coordinadas adecuadamente.

Una manera efectiva de lograr coordinación es imponer sobre los sistemas de agentes artificiales características propias de las organizaciones humanas. Estas organizaciones cuentan con patrones de comportamiento que regulan su funcionamiento. En particular, la especificación de normas, roles y relaciones restringe el comportamiento de los individuos, favoreciendo la concreción de las metas globales.

Nuestro trabajo parte del modelo de comportamiento para un agente BDI propuesto en [9] y lo extiende para incluir interacción a través de *diálogos colaborativos*. Cada individuo actúa restringido por las normas de la organización a la que pertenece e interactúa condicionado por la relación entre su rol y el de sus interlocutores.

La sección que sigue presenta alternativas para la especificación y asignación de roles. A continuación se introduce el concepto de norma social como un mecanismo para regular el comportamiento de un individuo responsable. La cuarta sección describe brevemente las características abstractas de un agente BDI dotado de razonamiento práctico y propone una alternativa basada en argumentación para representar el conocimiento y el mecanismo de planificación. La quinta sección extiende la alternativa propuesta para incluir roles, relaciones y una estructura normativa que regula la interacción entre los agentes. El algoritmo que modela el comportamiento de cada individuo en particular refleja su intervención en un diálogo colaborativo, tanto cuando actúa como iniciador como cuando interviene como colaborador. Por último se presentan algunas conclusiones y el trabajo futuro.

2 Roles y relaciones

En un contexto social el rol de un individuo determina el comportamiento esperado durante la interacción con otros miembros del grupo. En [11] se indica que los roles agrupan tipos de comportamiento en unidades significativas que contribuyen al logro de los objetivos del grupo. Las relaciones establecen vínculos entre los roles y permiten coordinar acciones, anticipar la evolución de la interacción y reducir el nivel de incertidumbre que provoca el dinamismo en el entorno. Una relación típica es la autoridad vertical o jerárquica.

La *especificación de los roles* en una organización artificial puede ser implícita o explícita. En el primer caso los roles se usan en forma intuitiva para analizar las cualidades de un sistema multiagente. En el segundo caso el concepto de rol toma mayor trascendencia porque brinda una estructura formal para construir sistemas multiagente para contextos dinámicos e impredecibles.

La *asignación de roles* puede ser fija o dinámica. El primer caso es el más simple y el rol de cada agente puede quedar establecido en el diseño de la aplicación. Cada miembro del sistema puede conocer su propio rol y el de los demás.

Una alternativa más flexible es que el rol de cada individuo pueda cambiar y deba ser considerado en el contexto de cada meta específica. Algunos miembros pueden estar dotados de la facultad de asignar roles determinando la estructura organizacional del sistema. La capacidad deliberativa de estos miembros probablemente sea más compleja que la de los demás, ya que requiere conocer las capacidades específicas de cada uno.

Una posición intermedia es que a cada agente se le asigne un rol en el momento en el que ingresa en el sistema. De este modo, el rol no queda fijado en el diseño, pero una vez que se establece, no cambia. Aun en este caso existen diferentes maneras de asignar un rol, entre ellas:

- *Imposición*: todo agente recibe un rol sin participar en la decisión. El objetivo puede ser, por ejemplo, que todos los miembros ocupen roles diferentes o por el contrario mantener varios agentes dentro de un mismo rol.
- *Aceptación sistemática*: todo agente es admitido en el sistema con el rol con el que desea ingresar.
- *Condicionada a un diálogo*: un agente que ingresa a un sistema expresa de alguna manera el rol que desea ocupar y si no es aceptado puede negociar hasta alcanzar un acuerdo que convenga a todos.
- *En función de las competencias*: en este esquema las habilidades y competencias de un agente van a determinar el rol o los roles que puede cumplir. Se requiere que la especificación de cada rol defina el conjunto de aptitudes que requiere y que la especificación de cada agente establezca el conjunto de habilidades y competencias que dispone.

Una propiedad fundamental es la *cardinalidad de los roles*, establece el número de agentes que deberían ocupar cada rol, como así también el número de roles que puede cumplir un individuo dado. La alternativa más estricta es que un agente ocupe exactamente un rol y un rol corresponda a un único agente. La más flexible es que cada rol pueda ser ocupado por un número no restringido de agentes, que a su vez pueden ocupar varios roles. Una posición intermedia, más compleja de modelar, es que algunos roles pueden tener restricciones específicas, por ejemplo puede haber roles de los cuales se exige que haya al menos un agente que ocupe esa función en cada momento dado o que haya a lo sumo uno asignado al rol. En particular, puede exigirse que exista un rol especializado, el administrador, entre cuyas funciones estará la admisión de nuevos miembros o el manejar los requerimientos de colaboración.

3 Las normas sociales

Las normas son fenómenos sociales que prescriben y proscriben patrones de comportamiento [12]. Un mismo individuo puede formar parte de diferentes grupos sociales, de modo que en cada uno deberá adaptarse a la estructura normativa correspondiente.

Las normas, así como también sucede con los roles, son elementos más o menos *estables* dentro de la vida social, por lo cual muchas veces pueden considerarse entidades estáticas. Sin embargo, las sociedades sufren transformaciones que las afectan. Estas influyen en el comportamiento social pero además son construidas a través de procesos sociales, basados en el comportamiento social. Esto es las normas son condiciones y resultados de procesos sociales [13].

En el diseño de la estructura normativa de una organización es importante considerar dos aspectos: la *perspectiva individual*, considerando como afectan las normas al mecanismo de razonamiento de los actores, y la *perspectiva institucional*, analizando el impacto de las normas sobre los procesos sociales que ocurren dentro de una organización.

Desde la perspectiva individual las normas pueden ser *restrictivas* o *impositivas*. Las normas restrictivas prohiben la ejecución de ciertas acciones en situaciones determinadas. Las normas impositivas exigen la ejecución de acciones en circunstancias establecidas.

Desde un enfoque institucional, algunas normas están codificadas en forma de *leyes* mientras que otras constituyen *convenciones*. La principal diferencia reside en que toda ley va a estar acompañada de algún mecanismo para detectar comportamiento ilegal y probablemente penalizarlo, en el caso de que se produzca. Las convenciones son lineamientos más débiles, pautas o guías que influyen en el comportamiento, sin restringirlo severamente.

La acciones de los individuos, grupos y sociedades van a estar también determinados por *principios* y *valores* que guiarán el comportamiento. Mientras que las normas son restricciones concretas que se aplican en circunstancias específicas, los principios y valores son lineamientos más generales que afectan a un espectro más amplio de situaciones. Los principios de una sociedad pueden ser brindar protección a todos sus miembros y en particular a los más vulnerables. Los valores podrían ser libertad y solidaridad. Los principios y valores pueden afectar la construcción de la estructura normativa.

Las normas, principios y valores guían el comportamiento a través de la generación de *obligaciones*. El término *responsabilidades* se utiliza normalmente como hiperónimo de principios, valores y normas. Una persona responsable es fiel a sus principios, respeta los valores del grupo al que perteneces, cumple con las normas que afectan a cada situación específica y se compromete con sus obligaciones.

En el diseño de una organización artificial las responsabilidades pueden quedar establecidas en forma implícita o explícita. En el primer caso las pueden estar embebidas en el *protocolo de interacción*, restringiendo las acciones y las intervenciones que cada agente puede realizar o en la *arquitectura* de cada agente individual, estableciendo qué prioridad asigna a cumplir con sus obligaciones, alcanzar sus metas y brindar colaboración.

Las normas pueden vincularse a la estructura de roles de maneras diferentes. Por un lado una norma implícita en el protocolo regula las interacciones posibles de acuerdo a la jerarquía de roles. Según la relación entre los roles que ocupan dos agentes en un momento determinado uno de ellos *podrá* exigir un comportamiento determinado. Como contrapartida el otro agente *deberá* actuar de acuerdo a lo requerido. Además a cada rol podrán asociarse normas explícitas de manera tal que los agentes que ocupan ese rol deben cumplirlas. La normas impositivas pueden ser controladas por un agente con un rol especifico que monitorea la ocurrencia de los eventos o pueden ser controladas por diferentes agentes.

4 Conocimiento y Acción

Un agente es una entidad computacional capaz de percibir, reaccionar y actuar en el entorno en el que se haya situado. Un agente racional está dotado además de algún nivel de *razonamiento práctico*, que le permite seleccionar una acción adecuada a partir del conocimiento de las metas y del entorno [7].

El razonamiento práctico involucra dos procesos fundamentales: *deliberación* y *razonamiento*

sobre medios y fines. La deliberación permite decidir **qué** metas van a perseguirse y requiere considerar un espectro de opciones alternativas, seleccionar algunas de ellas y comprometerse a cumplirlas. El razonamiento de medios y fines determina **cómo** van a alcanzarse las metas comprometidas, esto es permite construir un plan que le permita lograr sus intenciones.

El modelo de creencias, deseos e intenciones (BDI) basado en el razonamiento práctico, brinda los elementos esenciales para representar la actitud mental de un agente racional que actúa en un ambiente dinámico, sujeto a cambios bruscos y frecuentes [8] . El mecanismo deliberativo debe ser de algún modo *oportunista* en el sentido de que debe permitir seleccionar intenciones en función del beneficio que provoca, pero también considerando que el proceso de planificación pueda llegar a ser exitoso. Para ello es necesario considerar las características del entorno y de las habilidades del agente.

El *estado* de un agente BDI en un momento determinado se define en [9] como una terna $< \mathcal{C}, \mathcal{D}, I >$ tal que $\mathcal{C} \subseteq Cre$, $\mathcal{D} \subseteq Des$ e $I \subseteq Int$, donde Cre es el conjunto de todas las creencias posibles, Des es el conjunto de todos los deseos posibles e Int es el conjunto de todas las intenciones posibles. En el trabajo citado la función de generación de opciones mapea los conjuntos de creencias e intenciones previas en un nuevo conjunto de deseos.

Este trabajo presenta una alternativa a esta caracterización, modificando el proceso de revisión de deseos y el mecanismo deliberativo. El algoritmo que sigue presenta una variación para el modelo de comportamiento de un agente BDI propuesto en el trabajo citado antes:

$\mathcal{C} \leftarrow \mathcal{C}_0$
$\mathcal{D} \leftarrow \mathcal{D}_0$
while true **do**
  get next percept p
  $\mathcal{C} \leftarrow brf(\mathcal{C}, p)$
  $\mathcal{D} \leftarrow drf(p)$
  $iota \leftarrow deliber(\mathcal{C}, \mathcal{D})$
  $\Pi \leftarrow plan(\mathcal{C}, iota)$
  $execute(\Pi, \mathcal{C})$
end while

Los conjuntos $\mathcal{C}_0$ e $\mathcal{D}_0$ corresponden a las creencias y deseos iniciales. La función de revisión de creencias determina el conjunto de creencias a partir de la percepción actual y las creencias previas. En nuestra propuesta los deseos se actualizan también en cada ciclo a partir de la percepción actual. El mecanismo deliberativo genera una intención específica considerando exclusivamente los deseos y las creencias actuales.

La función plan retorna una secuencia de acciones a partir del conjunto de creencias y la intención comprometida, considerando por supuesto las habilidades del agente. La operación *execute* no abarca al plan completo, sino que luego de ejecutar cada acción individual se percibe y analiza la necesidad de reconsiderar intenciones o reformular el plan. La reconsideración de intenciones permite tomar ventaja de los cambios y aprovechar nuevas oportunidades.

El modelo BDI es abstracto, no especifica cómo se representa el conocimiento de un agente, ni propone un mecanismo deliberativo concreto o una estrategia de planificación específica. La selección de una intención puede realizarse de diferentes maneras, una de ellas es que todos los deseos tengan la misma importancia, otra, asociarle a cada una de ellos una *prioridad*.

En este trabajo el conjunto de metas que un agente percibe e incorpora a sus deseos se corresponde con las acciones que es capaz de ejecutar. Esto no garantiza que se pueda construir un plan ejecutable, el conjunto de creencias puede resultar insuficiente. Decimos entonces que el plan está incompleto. Cuando la ejecución de un plan termina exitosamente, la intención ha sido alcanzada y se transforma en una creencia.

A continuación se describe brevemente el formalismo adoptado para representar el conocimiento de un agente BDI y soportar planificación. Más adelante se propone una variación sobre al algoritmo que modela el comportamiento de un agente BDI con compromiso social.

4.1 Representación del conocimiento de un agente BDI

El modelo BDI brinda los elementos esenciales para representar la actitud mental de un agente racional que actúa en un ambiente dinámico, sujeto a cambios bruscos y frecuentes. Las creencias conforman el conocimiento del agente acerca del mundo. Sus deseos e intenciones se refieren al estado que desea alcanzar y representan sus motivaciones y compromisos. La capacidad efectora está constituida por el conjunto de acciones que es capaz de ejecutar.

Definition 4.1 [Agente Argumentativo BDI]
Un agente argumentativo BDI puede denotarse como una tupla $\mathbf{a} = \langle \mathcal{C}, \mathcal{D}, \iota, \Gamma \rangle$*, donde* $\mathcal{C}$ *representa el conjunto de creencias del agente,* $\mathcal{D}$ *sus deseos,* ι *la intención comprometida y* Γ *es el conjunto de acciones que es capaz de ejecutar.*

En este trabajo las creencias $\mathcal{C}$ están representadas por un Programa en Lógica Rebatible (Φ, Δ), donde Φ es un conjunto consistente de hechos, y Δ un conjunto de reglas rebatibles. La descripción detallada de DeLPpuede obtenerse en [2]. El conjunto de acciones Γ le permiten al agente cambiar el mundo de acuerdo a la propuesta presentada en [1]. Los deseos de un agente se representan como un conjunto de literales. Una intención es un literal específico, seleccionado dentro del conjunto de deseos como una meta comprometida.

Definition 4.2 [Acción] *Una acción A es una terna ordenada* $\langle \mathsf{P}, \mathsf{X}, \mathsf{C} \rangle$*, donde* P *es un conjunto de literales que constituyen las precondiciones de A,* X *es un conjunto consistente de literales que representan las consecuencias de ejecutar A, y* C *es un conjunto de restricciones de la forma not L, donde L es un literal. Denotaremos a las acciones como:*

$$\{X_1, \ldots, X_n\} \xleftarrow{A} \{P_1, \ldots, P_m\}, not\ \{C_1, \ldots, C_k\}$$

Notemos que not $\{C_1, \ldots, C_k\}$ *representa* $\{not\ C_1, \ldots, not\ C_k\}$*.*

Definition 4.3 [Acción aplicable] *Sea* $\mathcal{C} = (\Phi, \Delta)$ *las creencias de un agente y* Γ *el conjunto de acciones disponibles. Una acción A en* Γ*, es aplicable si toda precondición* P_i *en* P *tiene una garantía construída a partir de* (Φ, Δ) *y toda restricción* C_i *in* C *falla al intentar garantizarse.*

En este trabajo las normas restrictivas se modelan como precondiciones para la ejecución de las acciones. Así, una acción no será aplicable si en el estado actual del entorno la ejecución de dicha acción provoca la violación de una norma restrictiva.

Definition 4.4 [Efecto de una acción] *Sea* $\mathcal{C} = (\Phi, \Delta)$ *el conjunto de creencias de un agente* Γ *el conjunto de acciones disponibles para el agente. Sea A una acción aplicable en* Γ*Él efecto de ejecutar A es la revisión de* Φ *por* X*, i.e.* $\Phi^{*\mathsf{X}} = \Phi^{*\{X_1,\ldots,X_n\}}$*. La revisión consistirá de remover cualquier literal de* Φ *que es el complemento de cualquier literal en* X *y luego agregar* X *al conjunto resultante, esto es,* $\Phi^{*\mathsf{X}} = \Phi^{*\{X_1,\ldots,X_n\}} = (\Phi - \overline{\mathsf{X}}) \cup \mathsf{X}$*, donde* $\overline{\mathsf{X}}$ *representa el conjunto de los complementos de los miembros de* X*.*

En [1] se describe la interacción entre acciones y el formalismo de argumentación rebatible. Por un lado la argumentación rebatible se usa para testear las precondiciones y restricciones a través de la noción de garantía. Por otra parte, los agentes ejecutan acciones para cambiar el mundo de manera tal que sea posible obtener nuevas garantías.

4.2 Planificación basada en argumentación rebatible

Cuando un agente **a** adopta una intención ι, y ι no está garantizada por (Φ, Δ) el agente **a** busca una acción A en Γ que modifique las creencias de forma tal que pueda obtenerse un argumento sin derrotadores que soporte a ι. Si A no puede ejecutarse porque sus precondiciones no están garantizadas, **a** elabora una secuencia de acciones que le permitan establecerlas. La secuencia completa, incluyendo a A, conformará un plan para ι. Cada acción del plan puede modificar las creencias agregando literales que permitan construir nuevos argumentos. Al terminar la ejecución del plan la intención ι se agrega al conjunto de creencias $\mathcal{C}$.

La selección de acciones no es una tarea trivial y los problemas clásicos de planificación se reflejan en el esquema argumentativo. En una secuencia de acciones [A_1,A_2], A_2 puede ser aplicable de acuerdo al estado inicial Φ, pero no luego de que A_1 se ejecute. En efecto, las consecuencias de A_1 pueden modificar Φ de modo tal que las precondiciones de A_2 no estén garantizadas o una restricción si lo esté.

La ejecución de A_1 puede agregar literales que permitan construir nuevos derrotadores para las precondiciones de A_2 y eliminar literales que permiten construir garantías para las precondiciones de A_2. El criterio propuesto en [1] para seleccionar acciones es minimizar los cambios no deseados en las precondiciones y las restricciones.

Hemos indicado que en este trabajo un agente sólo adopta intenciones para las cuales puede construir al menos un plan. Esto es, existe al menos una acción entre cuyas poscondiciones está incluido el literal que corresponde a la intención. Sin embargo, es posible que las precondiciones de esta acción no se satisfagan considerando el conjunto de creencias del agente, ni este pueda elaborar un plan que le permita generarlas. En este caso el plan estará incompleto. Si el agente está aislado un plan incompleto no puede ejecutarse.

Cuando un agente forma parte de un sistema y sus capacidades resultan insuficientes para completar un plan, puede solicitar colaboración. Como contrapartida, sus acciones pueden interferir con los planes de los demás miembros. Si su actitud es cooperativa, brindará colaboración cuando reciba requerimientos y solicitará autorización para provocar cambios que puedan afectar a otros.

5 Conocimiento e Interacción

El plan de un agente individual se construye considerando únicamente sus creencias, intenciones y habilidades. Cuando un agente forma parte de un sistema, puede inteactuar con otros individuos y aprovechar sus creencias y sus habilidades. La interacción aumenta las posibilidades de alcanzar las metas comprometidas, pero exige como contrapartida considerar el impacto que el comportamiento social provoca sobre la representación del conocimiento, el mecanismo deliberativo, el proceso de planificación y el modelo de comportamiento de cada agente individual.

En este trabajo un sistema multiagente se concibe como organización artificial, formada por conjunto de agentes BDI colaborativos que interactúan de acuerdo a los roles que tienen asignados y restringidos por ciertas normas.

Definition 5.1 [Organización Artificial]
Una organización artificial se define como una tupla $\mathcal{V} = \langle \mathcal{A}, \mathcal{K}, \mathcal{G}, \mathcal{N}, \mathcal{R}, \mathcal{E} \rangle$*, donde* $\mathcal{A}$ *es el conjunto de identificadores de agentes que lo integran,* $\mathcal{K}$ *el conjunto de creencias globales,* $\mathcal{G}$ *el conjunto de deseos globales,* $\mathcal{N}$ *es el conjunto de normas globales,* $\mathcal{R}$ *es el conjunto de roles que van a asignarse a los miembros del grupo y* $\mathcal{E}$ *es una estructura jerárquica que especifica las relaciones entre los roles.*

Las creencias globales son aquellas que están compartidas por todos los miembros de la organización y para que uno de ellos pueda modificarlo se requiere autorización. Los deseos globales se representan a través de un conjunto de literales que constituyen las metas de la organización.

Durante el proceso perceptivo individual cada agente actualiza sus deseos individuales seleccionando entre los deseos globales aquellos para los cuales es capaz de elaborar planes. Los deseos y las creencias globales se modifican cuando se percibe el mundo y como consecuencia de la actividad de los agentes.

Una vez que un agente transforma un deseo individual en una intención, remueve el literal del conjunto de deseos global, intentando evitar que otro miembro del grupo adopte la misma meta. Evidentemente existe la posibilidad de que dos agentes incorporen el mismo deseo a su conjunto de deseos específico, lo seleccionen como una intención y elaboren un plan para alcanzarla. Si ambos llegan a tener éxito y ejecutan su plan, se habrá desperdiciado esfuerzo porque hubiera bastado con que uno de ellos lo hiciera.

5.1 La estructura de roles y relaciones

En este trabajo los roles existen dentro de una organización, independientemente de que estén instanciados o no. En un momento dado, puede producirse la situación de que ninguno de los miembros de un grupo pueda asignarse a un rol determinado y no por eso deja de existir dicho rol.

Definition 5.2 [**Roles**] *El conjunto de roles se define como* $\mathcal{R} = \{\rho_1, \rho_2, \ldots, \rho_n\}$. *Cada rol* ρ_i *dentro del conjunto* $\mathcal{R}$ *contiene al conjunto de* aptitudes *requeridas, donde cada aptitud va a estar representada a través de un literal.*

El rol de cada agente cambia dinámicamente dependiendo de las competencias que va adquiriendo. Las competencias están integradas por sus creencias actuales y todos aquellos planes que ha elaborado y ejecutado previamente. Un rol puede ser asignado a un agente si el conjunto de aptitudes que conforman el rol forma parte de sus competencias. Esto es, este está incluido en sus creencias o existen planes ya elaborados que le permiten cumplir con las funciones inherentes al rol.

Definition 5.3 [**Relaciones entre Roles**] *La estructura de relaciones entre los roles se define como* $\mathcal{E} \subseteq \mathcal{R} \times \mathcal{R}$ *y constituye un orden parcial satisfaciendo las propiedades reflexiva, antisimétrica y transitiva, Además, supondremos que* $\mathcal{E}$ *verifica la siguiente propiedad: si* $\rho_1, \rho_2 \in \mathcal{R}$ *entonces* $\exists \rho_3 \in \mathcal{R}$ *tal que* $\rho_1 \mathcal{E} \rho_3$ *y* $\rho_2 \mathcal{E} \rho_3$, *es decir, siempre existe el supremo entre dos roles.*

En el momento en que se inicia un diálogo colaborativo se asigna a cada participante el rol que corresponde y así queda determinada también la relación que unirá a los interlocutores durante todo el proceso. Esta relación va a restringir el tipo de locuciones que puede realizar cada uno.

Bajo esta concepción, en un grupo particular cada agente tendrá en un momento determinado un rol específico y entre cada par de agentes puede existir una relación establecida por sus roles. Esta relación condicionará la evolución del diálogo entre ambos.

5.2 La estructura normativa

En este trabajo, los valores y principios van a estar embebidos en el comportamiento de los agentes. La representación de normas restrictivas e impositivas es en cambio explícita, algunas en forma global afectando a toda la organización, otras más específicas en el modelo cognitivo de cada individuo.

Los *valores* fundamentales serán la sinceridad, honestidad y cooperación. La honestidad implica que los agentes van a respetar las normas de la organización y los roles asignados. Es decir, el comportamiento está regulado por las normas y los roles dentro de la estructura. Así, un agente honesto cumple con las normas salvo en aquellas situaciones en las cuales un superior lo autoriza a no hacerlo. Aun así solo algunas restricciones van a relajarse, las normas globales no van a ser violadas, lo cual implica que eventualmente el sistema puede bloquearse requiriendo intervención externa. La honestidad también provoca que ningún miembro de la organización modifique el conocimiento compartido sin autorización.

La sinceridad implica que las acciones se seleccionan a partir de las metas, pero se consideran solo aquellas cuyas precondiciones se satisfacen en el estado actual del mundo. Por lo tanto los agentes no actúan asumiendo creencias que no tienen.

La cooperación implica que todos los individuos estarán dispuestos a colaborar cuando reciban un requerimiento, intentando balancear sus metas propias con su intención de colaborar. El nivel de compromiso depende del rol.

El *principio* será la coordinación de acciones individuales a través de la interacción, regulada por la estructura de roles. Aunque los agentes están dotados de diferentes repertorios de acciones, todos responden a un mismo modelo de comportamiento colaborativo.

Las normas globales están asignadas a la organización en su conjunto, expresan condiciones que deben satisfacerse en todo momento y actúan simultáneamente como restrictivas e impositivas. En principio el conjunto de normas globales está incluido en el conjunto de creencias globales. Las normas son fijas, pero las creencias cambian como consecuencia de las acciones de los agentes y de los cambios externos en el entorno.

Las normas globales son restrictivas porque limitan el comportamiento de los agentes. Ningún agente puede ejecutar en condiciones normales una acción que viole una norma global, provocando que el conjunto de normas deje de estar incluido en el conjunto de creencias. Así, las acciones de un agente cuyas poscondiciones entran en conflicto con las normas globales, constituyen aptitudes que sólo se utilizarán en circunstancias excepcionales.

Las normas globales son además imposiciones porque si un cambio externo en el entorno afecta al conjunto de creencias, de manera tal que el conjunto de normas deja de estar incluido en él, los agentes quedan obligados a seleccionar esta norma como un deseo con máxima prioridad y elaborar un plan para volver a una situación normal.

5.3 El conocimiento de un agente BDI con compromiso social

Hemos considerado hasta el momento que el conocimiento de agente BDI incluye las creencias acerca del dominio en el que actúa, los deseos, su intención comprometida y el conjunto de acciones que conforman su capacidad efectora. Un agente con compromiso social respeta las normas globales y también puede tener asignadas algunas normas impositivas específicas que consideren sus habilidades:

Definition 5.4 [**Norma Impositiva**] *Una norma impositiva $\mathcal{I}$ es un par ordenado $\langle \mathsf{H}, \mathsf{S} \rangle$, donde H es el conjunto de literales que caracterizan al estado en el cual corresponde disparar a $\mathcal{I}$, y S es un plan, esto es la secuencia de acciones que impone ejecutar la norma.*

Las normas impositivas disparan la ejecución de una secuencia de acciones cuando H está contenido en las creencias actuales del agente. Notemos que la especificación de las normas impositivas es fija, lo que cambian son las creencias del agente como producto de la percepción de cambios externos en el entorno. Las normas impositivas caracterizan a situaciones no deseadas que en general deben superarse tan pronto como sea posible. De modo que la secuencia debería contener pocas acciones y al menos la primera no debería tener precondiciones.

La situación más simple es que el primer elemento del par sea un conjunto con un único literal y el segundo elemento del par sea una acción simple. En este caso la norma prescribe una acción específica ante un hecho determinado. Por ejemplo, si se detecta humo, activar la alarma.

Una alternativa a este modelo es que el segundo elemento del par no sea directamente una secuencia de acciones sino un conjunto de literales que caracterizan el estado que debería alcanzarse. En este caso el agente es el responsable de elaborar un plan que le permita alcanzar este estado. Para el ejemplo anterior, si se detecta humo se debe alcanzar un estado en el cual la alarma esté activa, pero no se indica si esto se logra con una acción simple o con una secuencia de acciones.

Extendemos ahora la definición propuesta antes para incluir *normas impositivas específicas*, *requerimientos de colaboración* y *competencias*.

Definition 5.5 [Agente Argumentativo BDI con capacidad Social]
Definimos el conocimiento de un agente argumentativo BDI con capacidad social, como una tupla $\mathcal{K} = \langle \mathcal{C}, \mathcal{D}, g, \Gamma, \mathcal{P}, \mathcal{I}_a \rangle$*, donde* $\mathcal{C}$ *representa las* creencias *del agente,* $\mathcal{D}$ *sus* deseos*,* g *es una* intención*,* Γ *el* conjunto de acciones *que es capaz de ejecutar,* $\mathcal{P}$ *el conjunto de competencias adquiridas y* $\mathcal{I}_a$ *el conjunto de imposiciones específicas.*

El conjunto de deseos está formado ahora por dos subconjuntos $\mathcal{F}$ y $\mathcal{Q}$, representando respectivamente las metas percibidas en los deseos globales y los requerimientos de colaboración de otros agentes.

El conjunto $\mathcal{P}$ está conformado por los planes que el agente ha construido y ejecutado previamente y aumentan su conocimiento. De este modo, cada plan ejecutado tiene una doble funcionalidad, por un lado le permite al agente alcanzar una meta, pero además aumenta su experiencia.

Cada nueva competencia que se agrega al conjunto puede pensarse como una acción compuesta y esto resulta particularmente útil cuando un agente enfrenta las mismas metas reiteradamente. La existencia de un plan previo no implica necesariamente que siempre se evite construir otro nuevo, porque en cada oportunidad las creencias pueden ser diferentes y entonces las precondiciones de un plan conocido pueden satisfacerse o no.

Es importante notar que las acciones involucradas en un plan ejecutado pueden tener como precondiciones a creencias que fueron obtenidas gracias a solicitudes de colaboración, que probablemente provocaron procesos de negociación entre dos o más agentes. La representación explícita de las competencias podría distinguir las acciones que el agente pudo realizar por sí mismo, de aquellas que provocaron la interacción con otros agentes. No consideramos significativo incluir dentro de las competencias aquellos planes que resultaron infructuosos y no llegaron a ejecutarse.

El conjunto $\mathcal{I}_a$de *normas impositivas específicas* indica lo que se **debe** hacer bajo ciertas circunstancias. Por ejemplo, si hay fuego cortar la luz y activar la alarma.

Las *normas restrictivas* indican lo que **no** se puede hacer, expresan prohibiciones. Como vimos algunas normas restrictivas se representan a través de un conjunto global que no debería ser modificado en circunstancia normales por ninguna acción. Cada agente respeta las normas globales. En circunstancias excepcionales, un agente con un rol superior puede autorizar a otro a violar una de estas restricciones. Otras restricciones son *específicas* de un agente en particular y se representan asignándole a cada acción de ese individuo un conjunto de precondiciones que representan restricciones para su ejecución. Notemos que si no existe un mecanismo para factorizar acciones, una misma restricción va a aparecer en cada agente que tenga la habilidad que corresponde a la acción.

Una norma global se transforma en imposición cuando un cambio externo provoca una modificación en las creencias inconsistente con la norma. Por ejemplo si una norma global establece que el piso tiene que estar seco y en determinado momento las creencias se modifican indicando que el piso no está seco, inmediatamente la meta *piso seco* se transforma en un deseo global con máxima prioridad. El agente que se comprometa a esta intención deberá elaborar un plan para alcanzarla.

Notemos que existe una diferencia importante entre las normas globales que se transforman en imposiciones y las normas impositivas específicas de un agente. Para estas últimas el individuo tiene un plan, *sabe* como actuar en determinadas circunstancias. En el caso de que una forma global genere una imposición, el individuo que la adopta tiene que buscar un modo de restaurar el estado en el cual se cumple con la restricción.

5.4 El comportamiento de un agente BDI con compromiso social

La participación en un contexto social modifica el comportamiento de cada individuo de una organización. Cada agente puede solicitar colaboración para alcanzar sus metas, pero también está comprometido a brindar ayuda cuando recibe un requerimiento.

La introducción de normas y roles afecta al comportamiento de cada miembro de la organización e implica modificar el algoritmo que modela este comportamiento. La percepción de deseos y la selección de intenciones se realiza considerando también las normas globales y específicas. Además aumenta la complejidad de la interacción porque debe considerar las restricciones impuestas por los roles.

Aunque en este modelo todos los agentes respetan un mismo algoritmo general, el proceso de selección de una intención y la planificación va a variar según el rol.

$\mathcal{C} \leftarrow \mathcal{C}_0$
$\mathcal{D} \leftarrow \mathcal{D}_0$
while true **do**
  Determinar el rol del agente
  Obtener una percepción p
  $\mathcal{C} \leftarrow brf(\mathcal{C}, p)$ {revisión de creencias}
  $\mathcal{D} \leftarrow grf(\mathcal{D}, p)$ {revisión de deseos y requerimientos}
  $iota \leftarrow deliber(\mathcal{C}, \mathcal{D})$
  $P \leftarrow plan(\Pi, iota)$
  if $iota \in \mathcal{Q}$ **then**
    $OK \leftarrow dialogue()$
  else
    if $incomplete(\Pi)$ **then**
      $OK \leftarrow request()$
    end if
  end if
  if OK **then**
    $execute(\Pi)$
  end if
end while

Los roles están asociados a aptitudes y en cada ciclo se asigna al agente el máximo rol dentro de la estructura de roles considerando que sus competencias puedan cubrir las aptitudes requeridas por el rol. Una vez que el rol ha sido asignado el agente percibe el entorno para actualizar creencias, deseos y requerimientos de colaboración.

Un agente solo percibe como metas literales que corresponden a las poscondiciones de sus acciones. Si una norma global se activa, un agente puede adoptarla como un deseo con máxima prioridad, en tanto su repertorio de acciones resulte adecuado para cumplir con ella. Cada individuo puede recibir requerimientos de colaboración aun cuando no se correspondan con sus capacidades, en este caso se declarará incompetente para atender la solicitud.

En este trabajo, la incorporación de roles provoca que los requerimiento de colaboración de los superiores tengan prioridad sobre los deseos percibidos en el mundo, dependiendo del nivel de exigencia del requerimiento. Notemos que en nuestro modelo el agente no recibe la orden de ejecutar una acción concreta, sino de llegar a un estado específico. Es el agente quien decide cómo hacerlo.

La elaboración de un plan es ahora más compleja porque es necesario considerar los conflictos que pueden producirse con las normas restrictivas. Esto es, si el agente está comprometido a respetar una norma, tendrá en principio inhibida cualquier acción cuya poscondición la viole.

Esta restricción tiene vigencia salvo en algunos casos excepcionales. Por ejemplo, si un agente elabora un plan para cumplir con una norma impositiva, puede violar una restrictiva, si no existe un plan alternativo que pueda realizarse sin violar las normas. Del mismo modo puede hacerlo, si recibe un requerimiento con un nivel alto de exigencia de un agente con un rol mayor que el de él mismo.

Como antes, la estrategia para elaborar un plan específico escapa a los alcances de este trabajo. Una alternativa es en primera instancia intentar buscar entre los planes elaborados previamente alguno cuyas precondiciones se satisfagan considerando las creencias actuales. Si existe podría seleccionarse inmediatamente. Si no existe un plan cuyas precondiciones se satisfagan a partir de las creencias individuales, es posible elaborar uno nuevo o completar el anterior. En ambos casos, el agente puede requerir colaboración. El mecanismo de selección podría aumentarse asignando un costo a cada plan, de modo que entre varios planes en condiciones similares, se considere el menor costo.

Una vez que el plan ha sido elaborado el agente puede interactuar con otros miembros de la organización si no dispone de todo el conocimiento necesario como para ejecutarlo como se propone en [5]. La interacción se produce tanto cuando el plan ha sido elaborado para atender un requerimiento de colaboración, como cuando se trata de un deseo propio pero se necesita colaboración. En el primer caso el agente ya está enfrascado en un diálogo, de modo que continúa interactuando en este caso para realizar una contrapropuesta que le permita obtener el conocimiento necesario para ejecutar su plan. En el segundo caso el agente inicia un diálogo solicitando colaboración.

6 Conclusiones y trabajo futuro

Uno de los mayores desafíos en el desarrollo de un sistema multiagente es lograr un comportamiento previsible ante situaciones imprevisibles en el entorno. Las normas de una organización permiten controlar de alguna manera la autonomía de los individuos en circunstancias convencionales y evitar desviaciones. La posibilidad de violar las normas en casos de emergencia brinda la flexibilidad necesaria para adaptarse a cambios bruscos en el entorno. En este trabajo las normas solo son vulneradas considerando la estructura de roles y el énfasis en el requerimiento, de modo que el comportamiento sigue siendo controlado.

El lenguaje de interacción debe permitir expresar el nivel de exigencia en un requerimiento ofreciendo un repertorio de performativas adecuado. La estructura de roles restringe las locuciones posibles, de modo que en cada diálogo el agente con un rol superior podrá usar un conjunto más amplio de performativas. El protocolo de interacción que modela las posibles formas en las que puede evolucionar un diálogo colaborativo, está entonces fuertemente ligado a la relación entre los roles. Nuestro trabajo futuro se concentrará en reformular el lenguaje de interacción y el protocolo de interacción descripto en [4] considerando la introducción de roles y normas presentada en este trabajo.

Una extensión posible sobre este trabajo es asignar normas específicas a los roles. Así la asignación de un rol se realiza considerando las aptitudes requeridas y determina las responsabilidades inherentes.

References

[1] *Actions, Planning, and Defeasible Reasoning.* Simari, G.R., García, A.J., Capobianco, M.. In Proc. of the 10th Int. Workshop on Non-Monotonic Reasoning, Canada, pp. 377-384, 2004.

[2] *Defeasible Logic Programming: An Argumentative Approach.* A.J., Simari, G.R., Theory and Practice of Logic Programming, 4(1), pp 95-138, 2004.

[3] Capobianco, M., Chesñevar, C., Simari, G. *Argumentation and the Dynamics of Warranted Beliefs in Changing Environments.*, Journal of Autonomous Agents and Multi-Agent Systems, 11, 127-151, 2005.

[4] Rueda S. V. , Martinez M.V. *Interaction among BDI Argumentative Agents: A dialogue games approach*. XI Congreso Argentino de Ciencias de la Computación (CACIC), pp. 955-966.

[5] Rueda S. V. , García A., Simari G. *Collaboration through Deliberative Dialogues*. JCST Vol. 7. No. 2., pp. 125–133, 2007

[6] Fagin, R., Halpern J., Moses, Y., Vardi, M. *Reasoning About Knowledge*. The MIT Press, Cambridge, MA 1995.

[7] Bratman, M.E., Israel D.J., Pollack, M.E. *Plans and Resource-Bounded Practical Reasoning*. Computational Intelligence 4:349-355, 1988.

[8] Rao, A. S., and Georgeff M. P. *BDI Agents: From Theory to Practice*. Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), pp. 312-319, USA, 1995.

[9] Wooldridge, M. *Intelligent Agents*, The MIT Press, 1999.

[10] Huhns, M., Stephens, L. *Multiagent Systems and Societies of Agents*, The MIT Press, 1999.

[11] Partsakoulakis I., Vouros G. *Roles in Mas: Managing the Complexity of Tasks and Environments*. In *An Application Science for Multi-Agent Systems*, T. Wagner Ed., Springer-Verlag, pp. 133-154, 2004.

[12] Vázquez -Salceda J., Aldewereld H., Dignum F. *Norms in Multiagent Systems: Some Implementation Guidelines*. Second European Workshop on Multi-Agent Systems, pp. 737–748, Barcelona, 2004.

[13] Köhler, M. , Moldt, D., and Rölke, H. *A Discussion of Social Norms with Respect to the Micro-Macro Link*. In G. Lindemann, D. Moldt, and M. Paolucci, Eds., 2nd Int. Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'03), 2003. In connection with the Ninth ICAIL.

[14] Lopez y Lopez, F., Luck, M., and, d'Inverno, M., *A Framework for Norm-Based Inter-Agent Dependence*. In 3rd Mexican International Conference on Computer Science, pp. 31-40, 2001.

Utilización de Algoritmos Evolucionarios para Asignación de Recursos

L. Osorio[1], **E. Fernández**[1,2], **H. Merlino**[1,2], **D. Rodríguez**[1], **P. Britos**[1,2], **R. García-Martínez**[1,2]
[1]Centro de Ingeniería de Software e Ingeniería del Conocimiento. Escuela de Postgrado. Instituto Tecnológico de Buenos Aires
[2]Laboratorio de Sistemas Inteligentes. Facultad de Ingeniería. Universidad de Buenos Aires.
rgm@itba.edu.ar

Resumen

Los algoritmos Genéticos, y su generalización: los Algoritmos Evolucionarios; presentan una alternativa para solucionar problemas de optimización complejos. Si bien entenderlos es sumamente sencillo, debido a que se asemejan a los procesos naturales que poseen las especies para su supervivencia, aplicarlos a un problema real implica la construcción de un sistema de alta complejidad. El presente trabajo contiene el estudio, desarrollo, implementación y prueba de un sistema que aplica Algoritmos Evolucionarios para resolver el problema de la asignación de recursos humanos a tareas vinculadas a la construcción de cañerías para fluidos relacionados con el petróleo.

Palabras claves: algoritmos genéticos, logística, asignación de recursos humanos.

Abstract

Genetic algorithms and its generalization: evolutionary algorithms present an alternative to solve complex problems of optimization. If it is easy to understand them, because they are similar to the natural processes that the species possess for its survival, to apply them to a real problem implies the construction of a system of high complexity. This work presents the study, development, implementation, and testing of a system that applies evolutionary algorithms, to solve the problem about assigning human resources to tasks linked to the construction of pipings for fluids related to oil.

Key words: genetic algorithms, logistics, human resources assignment.

1- Introducción

En presente proyecto fue desarrollado para solucionar los problemas de asignación de recursos humanos a proyectos de construcción de gasoductos, oleoductos, acueductos y montaje prefabricado de cañerías en plantas de procesamiento relacionadas con la actividad petrolera. El mismo será utilizado por la empresa GLP en la provincia de Chubut.

En la actualidad, en la empresa GLP, el problema de la asignación de recursos comienza con el inicio de un emprendimiento por encargo de un cliente. Luego de la definición del emprendimiento la persona que se encarga de la administración y dirección de proyectos realiza la lista de tareas necesarias. Cada tarea involucra una determinada cantidad y tipos de recursos humanos para ser llevada a cabo. Luego de confeccionada la lista de tareas procede a realizar la búsqueda de los individuos que van a formar parte del emprendimiento; para ello tiene que acceder a un archivo de fichas individuales de personas que pueden ser contratadas. Esta búsqueda consiste en determinar la cantidad y los tipos de individuos que mejor se adapten a las tareas requeridas; pero a la vez debe tener en cuenta las horas ociosas. Las horas ociosas se producen por tiempos en donde algunos individuos podrían no tener asignada ninguna tarea durante la ejecución del proyecto. En la figura 1, a modo de ejemplo se muestra el armado de una tubería de Alta Presión:

Figura 1- Armado de uan tubería de Alta Presión

Para optimizar el proceso de asignación de recursos a las distintas tareas se desarrollo sistema que aplicando Algoritmos Evolucionarios[1] optimisa la cantidad de horas ociosas (obteniendo la dotación de personal que menor cantidad de horas ociosas produzca) en función del personal disponible para la obra y la cantidad y tipo de tareas a desarrollar.

2 Estado de la Cuestión

2.1 Tareas y Recursos

Cuando a GLP se le encomienda un nuevo emprendimiento necesita organizar sus recursos humanos en función de las tareas que lo componen. Para ello la persona encargada de la asignación enumera la lista de tareas necesarias junto con los recursos humanos que cada una de estas tarea necesita. Luego comienza a verificar si en la lista de personal de empresa se encuentran las personas que pueden llevar a cabo cada una de las tareas; si existen tareas que no están cubiertas por el personal existente en GLP se inicia el proceso de búsqueda y contratación. Una vez que todas los puestos están asignados puede lanzarse la ejecución del emprendimiento. Este ultimo paso finaliza la tarea de asignación de recursos humanos. A continuación se definen los tipos de tareas, los tipos

de recursos humanos y las caraterísticas que hacen a los ditstintos recursos humanos mas o menos aptos para realizar las tareas.

- Tipos de Tareas:

  En la tabla 1 se enumeran las diferentes tareas que se pueden llegar a realizar en un emprendimiento de GLP.

Código	Tareas
T1AT	Análisis de terreno
T2AR	Asignación de recursos
T3DC	Desfile de cañerías
T4TP	Trazado de pista

Código	Tareas
T5SC	Soldadura de cañería
T6EC	Enterrado de cañería
T7PH	Prueba hidráulica
T8ES	Envío de Scraper

Tabla 1- Lista de tipos de tareas

- Recursos Humanos:

  En la tabla 2 se enumeran a todos los tipos de recursos humanos que pueden ser contratados por GLP.

Código	Descripción
H01JH	Jefe de obra
H02SM	Supervisor de movimientos de suelo
H03SS	Supervisor de soldadura
H04JS	Jefe de soldadura
H05JM	Jefe de movimientos de suelo
H06JC	Jefe de control de calidad
H07RA	Responsable de seguridad higiene y medio ambiente

Código	Descripción
H08TP	Topografos
H09SO	Soldadores
H10AM	Amoladores
H11AY	Ayudantes
H12CV	Choferes de equipos viales
H13CP	Choferes de transporte de personal

Tabla 2- Lista de tipos de recursos humanos

- Caraterísticas de los recursos

  En la tabla 3 se enumeran las diferentes carácterísticas que pueden tener los recursos humanos contratados por GLP.

Código	Descripción	Sub categorías
Capacidad	Es la experiencia que posee el individuo para realizar la tarea asignada.	excelente
		muy capaz
		capaz
		nuevo
Edad	Es la edad que posee el individuo.	senior
		semi senior
		junior
Antigüedad	Es el tiempo que posee el individuo trabajando en la empresa; o es conocido por la persona que realiza la asignación. La antigüedad se subdivide hasta 10 años.	Más de 10 años
		Entre 8 y 10 años
		Entre 6 y 8 años
		Entre 4 y 6 años

Código	Descripción	Sub categorías
		Entre 2 y 4 años
		Hasta 2 años
Liderazgo	Es la capacidad que posee el individuo de liderar un grupo de personas.	Líder natural
		Líder natural en cambio
		Líder en pequeños grupos
		Líder en tiempo de prueba
		No está definido como líder
		No es líder

Tabla 3- Lista de características de los individuos

2.2- Algoritmos Evolucionarios

Los Algoritmos Evolucionarios son especialmente adecuados para resolver problemas difíciles de optimización. Según la definición [1]; [2]; [3]; [4] los algoritmos evolutivos son un concepto general adaptable para la resolución de problemas, y no una colección de algoritmos relacionados y listos para ser usados.
Los Algoritmos Evolucionarios tienden a lograr soluciones cada vez más eficientes debido a dos acciones: la primera es que *selecciona* las mejores poblaciones para la siguiente generación. La segunda acción es la *combinación* (o migración) de las mejores poblaciones para crear otras poblaciones.
Estos algoritmos evolucionarios en general, datan desde 1950, pero en las últimas décadas emergieron las tres variantes más importantes: Programación Evolucionaria (PE), Estrategia Evolucionaria (EE) y Algoritmos Genéticos (AG). Estas variantes implementan algoritmos evolucionarios de diferente manera.
La estructura de un algoritmo evolucionario [5]; [6] se observa en la figura 2. Este algoritmo mantiene una población de estructuras que renueva de acuerdo a las reglas de selección, migración, re combinación y mutación. El ciclo repetitivo finaliza cuando se cumple la condición de parada.

```
t = 0
generar lista de poblaciones iniciales LP(t)
evaluar lista de poblaciones LP(t)
Hasta (condición de parada)
   t = t + 1
   seleccionar m mejores poblaciones de la lista de poblaciones LP(t)
   realizar migraciones entre poblaciones
   recombinar en cada población de la lista de poblaciones LP(t)
   mutar en cada población de la lista de poblaciones LP(t)
   evaluar cada población de la lista de poblaciones LP(t)
Fin Hasta
```

Figura 2- Estructura genérica de un Algoritmo Evolucionario

Existen dos elementos principales para definir un algoritmo evolutivo: los operadores y los criterios de parada. A continuación se realiza una breve definición de cada uno de ellos [7]; [8]:

- Operadores

  A continuación se describen los cuatro operadores básicos usados por los "Algoritmos Ebolucionarios":

  - *Selección:*

    Por medio de este operador se selecciona a las mejores poblaciones de individuos que van a ser tenidas en cuenta por el resto de los operadores. El operador de selección es uno de los operadores que utiliza la función de aptitud.

  - *Migración:*

    La migración consiste en hacer que algunos individuos emigren de su población original y se trasladen a otra población.

  - *Cruza:*

    La cruza simula la relación sexual de los individuos naturales. En las poblaciones este operador genera nuevos individuos. En general se selecciona dos individuos y producen de uno a dos hijos.

  - *Mutación:*

    La mutación consiste en pequeños cambios en las características de los individuos. Este operador se aplica a un porcentaje pequeño de la población.

- Criterios de parada

  Por cada ciclo de corrida se evalúa si se continua o no con un nuevo ciclo, o generación. Esa decisión se establece por medio de los llamados criterios de parada. A modo de ejemplo se pueden citar: el número de generaciones generadas, el nivel de la función de evaluación, o cuando las poblaciones no pueden alcanzar mejores resultados.

3- Definición del Problema

Actualmente la asignación de recursos humanos a tareas, en la empresa GLP, se realiza en forma manual. La persona encargada de realizarla dicha labor toma cada tarea del emprendimiento y verifica si puede ser cubierta con el personal idóneo; calcula el tiempo aproximado de inicio y finalización; y visualiza cuales de las mismas se puede ejecutar en paralelo a otra tarea del mismo emprendimiento. Esta forma de trabajar insume una gran cantidad de tiempo y la optimización de los costos no es la ideal, ya que no es tarea sencilla optimizar de forma manual la cantidad de horas ociosas cuando la asignación de individuos a las distintas tareas no solo se hacen en función se su especialidad, sino, que se debe analizar cuan idóneo es, cuantos tipos diferentes de tareas puede realizar (por ejemplo un "ayudante" puede también ser "chofer") y como se vincula esto con las tareas factibles de ser desarrolladas en paralelo.

4- Solución Propuesta

4.1- Definición del Sistema

El objetivo principal de esta solución es crear diferentes poblaciones de individuos disponibles para realizar las tareas de un emprendimiento específico. El proceso comienza tomando poblaciones

iniciales para luego aplicarles operadores de selección, migración, cruza y mutación; durante un ciclo determinado de iteraciones, con el fin de crear nuevas poblaciones que mejor se ajusten a dichas tareas.
La arquitectura del sistema, ver figura 3, esta compuesta por dos módulos: **PA01** y **PA02**; dos bases de datos **poblaciones** y **tareas;** más un reporte de salida **asignación**. Donde mientras el primerro de los módulos se encarga de la administración de las tareas y poblaciones, el segundo es el encargado de hacer correr las diferentes poblaciones.

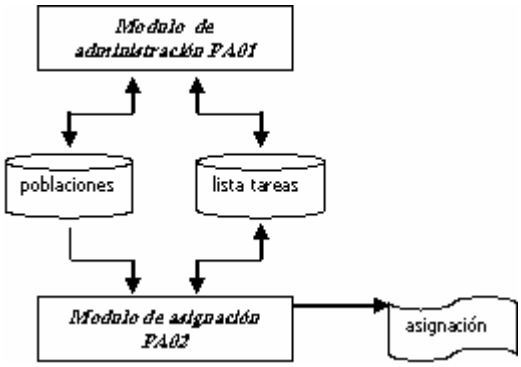


Figura 3- Arquitectura del sistema

En la figura 4 se describe el esquema de funcionamiento del sistema, en la misma se pueden observar dos ciclos anidados; el primer ciclo es el que se encarga de tomar una población de individuos, ejecutar la asignación de tareas y hacer correr el tiempo para ver si con dicha población se pueden ejecutar todas las tareas del emprendimiento; el segundo ciclo es el que posee el Algoritmo Evolucionario, este último ciclo contiene los operadores, función de aptitud y condición de parada.

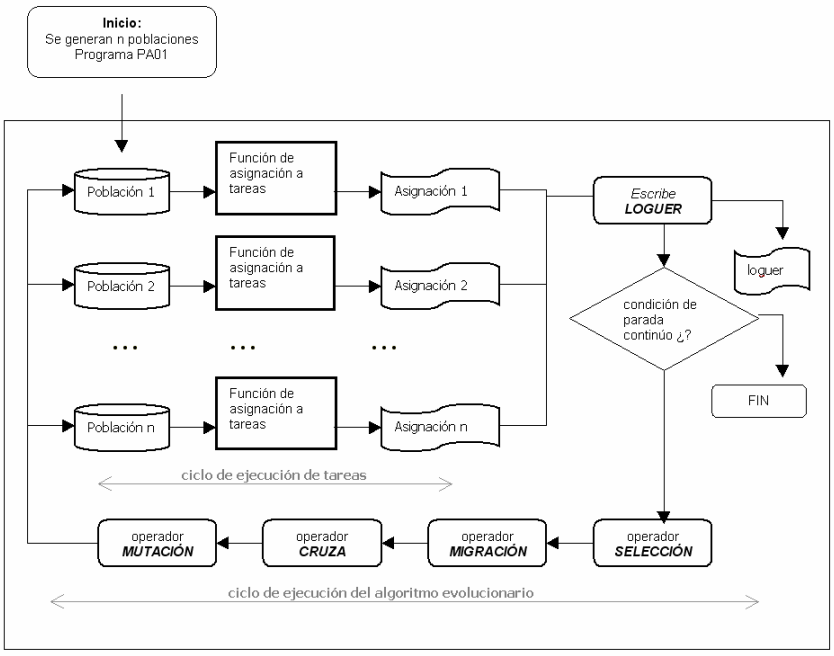


Figura 4- Esquema de funcionamiento del programa de asignación

Para optimizare las diferentes poblaciones, se utilizará la siguiente función de aptitud basada en horas ociosas:

$$total\_horas\_ociosas(p) = \sum_{i=1}^{m} horas\_ociosas(R_i)$$

Donde:
p: población
m: cantidad de individuos de la población p
Ri: individuo i de la población P
horas_ociosas(Ri): es la cantidad de horas ociosas que obtuvo el individuo Ri en la ejecución de las tareas.

4.2- Interfaz de la herramienta

A continuación se describen las pantallas principales que posee el sistema de asignación: en la figura 5 se detalla la pantalla del módulo pa01 encargado de configurar las poblaciones y emprendimientos; en la figura 6 se detalla la pantalla del módulo pa02 que posee los ciclos de ejecución que permiten la asignación de recursos humanos a tareas.

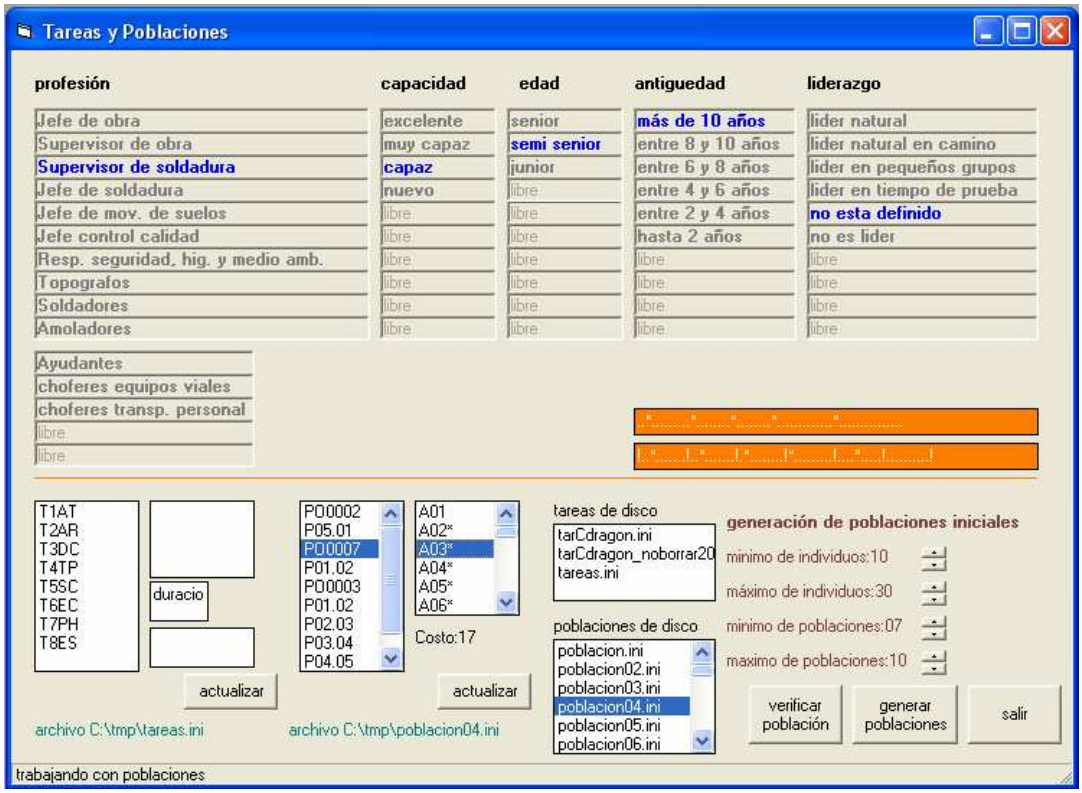


Figura 5- Pantalla del programa PA01 administración de poblaciones y tareas

Mediante la pantalla pa01 se pueden generar diferentes poblaciones cuyas cantidades y diferentes perfiles de personas tambien se generan al azar. Además permite configurar cada una de las tareas de un proyecto.

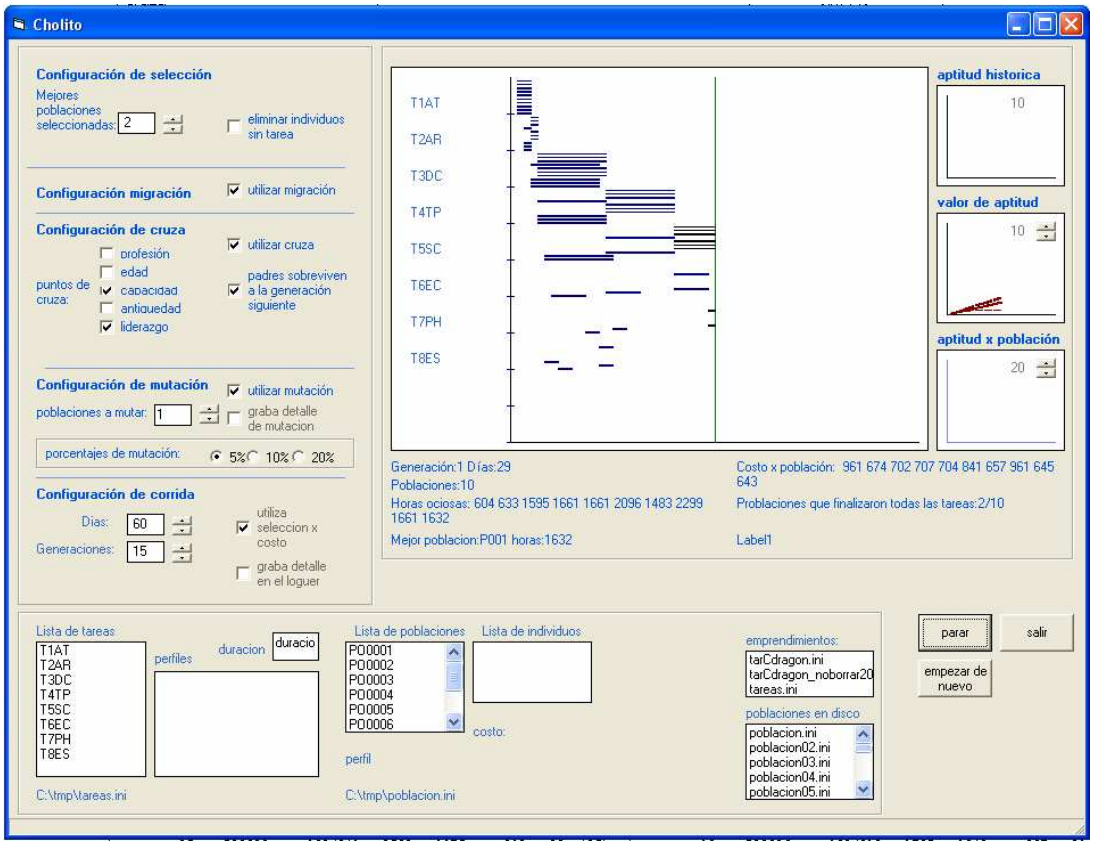


Figura 6- Pantalla del programa PA02 Asignación de recursos humanos a tareas

Mediante esta pantalla pa02 se pueden seleccionar diferentes poblaciones y proyectos; setear la cantidad de días y lanzar el proceso de asignación. En la misma pantalla se visualiza como día a día se desarrolla la ejecución del proceso de asignación.

5- Demostración de la solución

La figura 7 se muestra un estracto del informe generado por el sistema, en el mismo se observan la segunda y tercera generación. En la segunda generación las poblaciones po0002 y po0007 obtuvieron la menor cantidad de horas ociosas y se observa que las mismas aparecen en la tercer generación; es decir el algoritmo del programa las reservó para la generación siguiente. En otras palabras, el algoritmo evolutivo una vez que encuentra una mejor solución trata de preservarla para la siguiente generación; es decir trata cada vez de generar, al menos, iguales o mejores soluciones.
En la misma figura, en la tercer generación, aparece una nueva población p05.01 que obtuvo la menor cantidad de horas ociosas; esta población fue creada por el algoritmo combinando las poblaciones po0001 y po0002, de la generación anterior. De esta forma el algoritmo genético presupone que si combina de alguna las mejores poblaciones va a encontrar mejores soluciones.

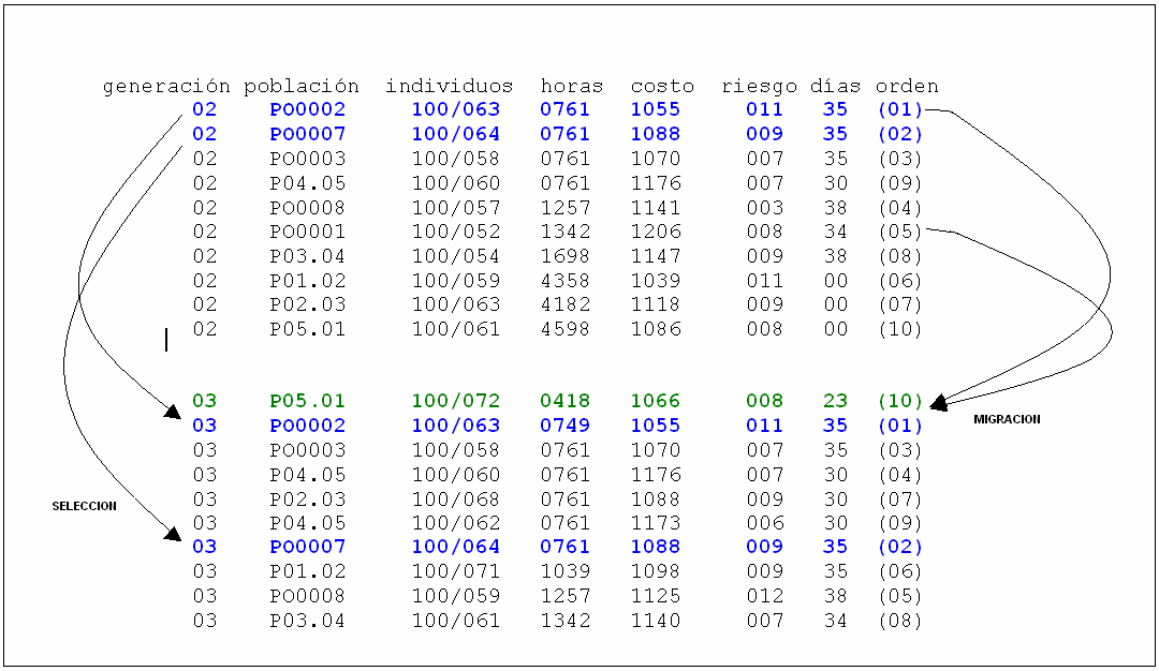


Figura 7- Loguer de prueba de Validación

Desde otro punto de vista el programa de asignación también podría seleccionar los mejores individuos y preservalos debido a que puede distinguir entre individuos que mejor se ajusten a las diferentes tareas y en función a ello seleccionarlos para el procesamiento de cruza. Esta capacidad no fue explotada en la versión del programa de asignación utilizado para este proyecto; solo se utilizó dos indicadores que calificaban de alguna forma a los individuos: su costo y su riezgo. El costo es una función aritmética que da mayor peso si el individuo tiene mejores calificaciones en su perfil; el riezgo es otra función aritmética que mide cuanto se acerca el individuo al perfil de la tarea; cuando más se ajueste el riezgo es menor.
A contiinuación en la figura 8 se presenta un gráfico que describe como varía la cantidad de horas osiosas de una genaración a otra dentro del proceso de asignación. Como puede verse al inicio del proceso la cantidad de horas osiosas alcanza casi las 100 horas y al finalizar el proceso este valor deciende casi al 50%.

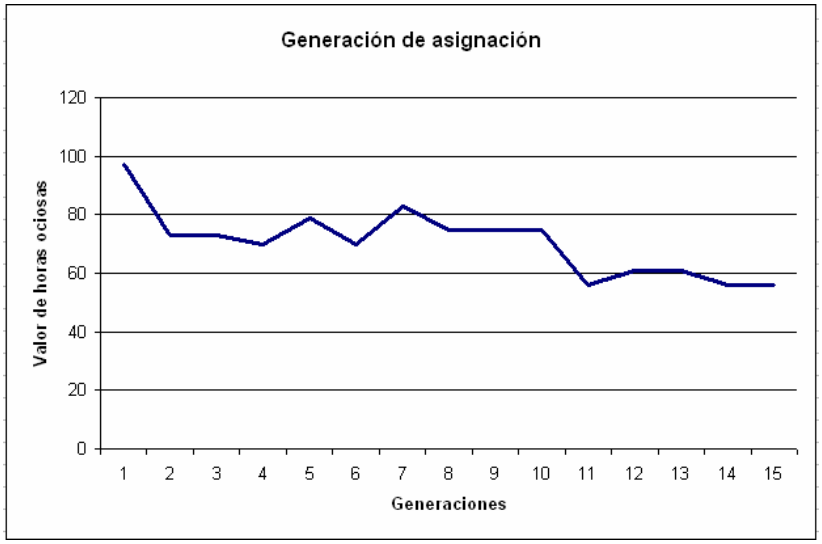


Figura 8- Medición de la cantidad de horas ociosas por generación

7- Conclusiones

Mediante este proyecto se ha podido corroborar la potencia de los algoritmos evolutivos para resolver problemas de optimización de asignación de recursos humanos a tareas: Se ha verificado que los Algoritmo Evolucionarios del módulo PA02 logra optimizar las poblaciones iniciales.

Por otro lado se ha podido corroborar los siguientes aspectos teóricos:

- Los algoritmos evolutivos logran mejores soluciones por que en ellos predominan los mejores individuos.
- El módulo PA02 pudo mantener los mejores individuos y poblaciones completas de una generación a otra.
- Los operadores de migración y cruza tuvieran mayor incidencia en la solución, que el operador de mutación (este concepto también se encuentra en la teoría propuesta por la cátedra).

Por último el módulo PA02 representa una mejora en cuanto que libera a la persona que realiza hoy la asignación de recursos en GLP de la cantidad y tipo de recursos humanos que se necesitan para realizar un emprendimiento.

8- Futuras líneas de investigación

Al presente sistema se le podrían realizar las siguientes mejoras:

- Utilización de operadores dinámicos, esto implica que los operadores se modifiquen durante la ejecución del proceso, por ejemplo, el operador de mutación podría incrementar o reducir su porcentaje de incidencia en función al resultado que se va obteniendo durante la corrida del sistema.
- Modificación de la función de aptitud, el sistema podría tener otros componentes y funcionar en base a otras variables. Por ejemplo la función de aptitud podría tener en cuenta horas ociosas y tiempo de ejecución total del emprendimiento.
- Modificar el operador de selección, el cual actualmente cuando encuentra dos individuos que cumplen con el perfil, decide con cual quedarse en función al costo del individuo, esto podría modificarse incorporando un valor de riesgo que es la medida de ajuste de un individuo al perfil de la tarea.
- Expandir la utilización del sistema a **otras áreas**, por ejemplo en la construcción de viviendas, organización de eventos sociales o construcción e implementación de sistemas de información.

9-Referencias

[1] García Martínez,R. *”Algoritmos Genéticos”*. Centro de Ingeniería del Software e Ingeniería del Conocimiento. Instituto Tecnológico Buenos Aires.

[2] Erick Cantú-Paz "A Summary of Research on Parallel Genetic Algorithms" IlliGAL Report No. 95007, University of Illinois July 1995

[3] Kenneth A. De Jong, Williatn M. Spears "A Formal Analysis of the Role of Multi-Point Crossover in Genetic Algorithms" Annals of Mathematics and Artificial Intelligence Joumal, Vol 5, No. 1, 1992

[4] David E. Goldberg "Genetic Algorithms in Search, Optimization, and Machine Learning" Addison-Wesley Publishing Company, 1989

[5] David E. Goldberg "Genetic and Evolutionary Algorithms Come of Age Communications of the ACM, Vol. 37, No. 3, March 1994

[6] David E. Goldberg, Brad L. Milier "Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise" IlliGAL Report No. 95009, University of Illinois, 1995

[7] John H. Holland "Schemata" GA-List, GA Vol. 8 No. 26,1994 [MIL/95] Brad L. Miller, David E. Goldberg "Genetic Algoriduns, Toumwnent Selection, and the Effects of Noise" IlliGAL Report No. 95006, University of Illinois, 1995

[8] William M. Spears "Crossover or Mutation?"Proceedings of the Foundations of Genetic Algoritluns Workshop Vail, Colorado, pag. 221- 237, July 1992

Utilización de Excepciones para Implementar Predicados Opacos en Técnicas de Ofuscación de Código Intermedio

Daniel Dolz
daniel_jose_dolz@yahoo.com.ar

Gerardo Parra
gparra@uncoma.edu.ar

Departamento de Ciencias de la Computación
Facultad de Economía y Administración
Universidad Nacional del Comahue
Buenos Aires 1400 - 8300 Neuquén - Argentina
Tel/Fax (54) (299) 4490312/313

Abstract

Microsoft's .NET Framework, and JAVA platforms, are based in a just-in-time compilation philosophy. Software developed using these technologies are executed in a hardware independent framework, which provides a full object-oriented environment, and in some cases allows the interaction of several components written in different programming languages.
This flexibility is achieved by compiling into an intermediate code which is platform independent. Java is compiled into ByteCode, and Microsoft .NET programs are compiled into MSIL (Microsoft Intermediate Code). However, this flexibility comes with a price. It is really easy, with tools available for free in the web, to decompile this intermediate code and obtain a working, readable version of the original source code.
Of all techniques developers can use to protect their intellectual property, obfuscation is the most accepted and commercially available one.
In the present work, we propose the use of try-catch mechanisms available in .NET as a way to improve the quality of one of the building blocks of obfuscation: opaques predicates.

Keywords: Obfuscation. Obfuscation Transformation. Opaque Predicates.

Resumen

La plataforma .NET de Microsoft se basa en una filosofía de *just-in-time compilation* (compilación bajo demanda al momento de la ejecución). Los programas desarrollados de esta manera se ejecutan en un entorno o framework independiente de la plataforma, basado en objetos y, en algunos casos, permitiendo que interactúen componentes desarrollados en distintos lenguajes de programación.
La clave de esta flexibilidad se da en que, tanto en JAVA como en las plataformas .NET, la compilación resulta en un código intermedio, independiente de la plataforma (bytecode y MSIL respectivamente). Sin embargo, dicha flexibilidad tiene un costo. Hoy en día, y utilizando herramientas gratuitas que pueden descargarse desde Internet, es sumamente fácil

aplicar tecnologías de ingeniería inversa a las dos plataformas de desarrollo más populares: JAVA y .NET.
De todas las técnicas que los desarrolladores pueden utilizar para proteger su propiedad intelectual, la ofuscación es la técnica más aceptada y de hecho, es la única utilizada comercialmente.
En este trabajo, proponemos el uso de los mecanismos de excepción (bloques try-catch) que brinda la plataforma .NET como una manera de mejorar la calidad de uno de los bloques básicos de la ofuscación, los predicados opacos.

PALABRAS CLAVES: Ofuscadores. Código Intermedio. Transformaciones de Ofuscación. Predicados Opacos.

1 Introducción

Tanto las aplicaciones JAVA, como aquellas desarrolladas para ejecutarse sobre cualquiera de las versiones del Framework de .NET, poseen la vulnerabilidad de que, con herramientas gratuitas y con solo conocimientos básicos de informática, es posible para cualquier persona que posea los distribuíbles de la aplicación obtener, de manera completa y con solo ligeras variaciones, el código fuente original completo de la aplicación.
La apropiación del código fuente de un software desarrollado por una organización en manos de personal no autorizado y con intenciones evidentemente ilegales, podría tener las siguientes consecuencias:

- Pérdida a manos de la competencia del dinero invertido en I+D[1]. La competencia puede, de forma desleal, lanzar al mercado el mismo producto, con un "lavado de cara" y aprovechando la inversión de la organización original.
- Una organización competidora podría descubrir fallas en el producto y utilizarlas en su beneficio.
- En el caso puntual de los algoritmos de encriptación modernos, cuya seguridad está basada en la existencia de una clave desconocida y no de un algoritmo en particular, el acceso al algoritmo por parte de manos malintencionadas podría servir para, previa modificación de los mismos, intentar ataques de fuerza bruta contra los datos cifrados.
- El acceso al código fuente de una aplicación facilita el "crackeo" de sistemas anti piratería, como ser la registración de software mediante *keys, expiration dates, hardlocks,* etc.
- Un empleado con conocimientos de informática podría, descompilar la aplicación de gestión administrativa de la empresa, modificar los strings de selección a las bases de datos de manera de eliminar restricciones y filtros, recompilarla, ejecutarla, y obtener acceso irrestricto a la base de datos de clientes. Esto podría resultar en la pérdida de valiosos secretos comerciales.

Lo anterior indica problemas económicos, pero algunos gobiernos como el de los EEUU identifican al problema como de seguridad nacional [4]. En el ámbito privado, se sabe que el 75% de las empresas Fortune 500[11] utilizan de una manera u otra el paquete de desarrollo Microsoft Visual Studio 2005[5]. Viendo que las principales amenazas a la seguridad provienen, hoy por hoy, no de agentes externos a las organizaciones sino de elementos internos de la misma con acceso a los recursos de la empresa desde adentro (empleados, personal contratado, consultores, etc.), el problema no es menor [6].

[1] I+D: Investigación y Desarrollo

Nuestra línea de trabajo, iniciada en [12], muestra a las técnicas de ofuscación como la rama de la seguridad informática que puede brindar un nivel de protección superior al de las alternativas existentes. En este artículo mostramos cómo, mediante la utilización de excepciones, es posible incrementar de manera notoria la calidad de los predicados opacos, mejorando en consecuencia la calidad de la ofuscación y por ende la protección de la propiedad intelectual.

La estructura del trabajo es la siguiente. A continuación, se brinda una descripción introductoria de la ofuscación y de sus conceptos claves. Luego, se profundiza en las construcciones denominadas predicados opacos. En la sección 4, presentamos el aporte de este trabajo. Se describen las técnicas de generación de predicados opacos más avanzadas y se presenta la innovación de reemplazar el uso de sentencias de salto condicional por bloques del tipo *try-catch*. Finalmente, en la sección 5, se reportan las conclusiones y se presentan algunas líneas de trabajo futuro.

2 Ofuscación, conceptos claves

En términos generales, se entiende por ofuscar un código fuente o un código intermedio, un proceso mediante el cual se transforma utilizando diversos algoritmos de reescritura, un código perfectamente legible y entendible por una persona en otro de funcionalidad equivalente en un ciento por ciento, pero, en términos ideales, totalmente ilegible e incomprensible para un lector humano.
En la figura 1 se muestra, de manera esquemática, el proceso y concepto de aplicación de ofuscación de código intermedio.

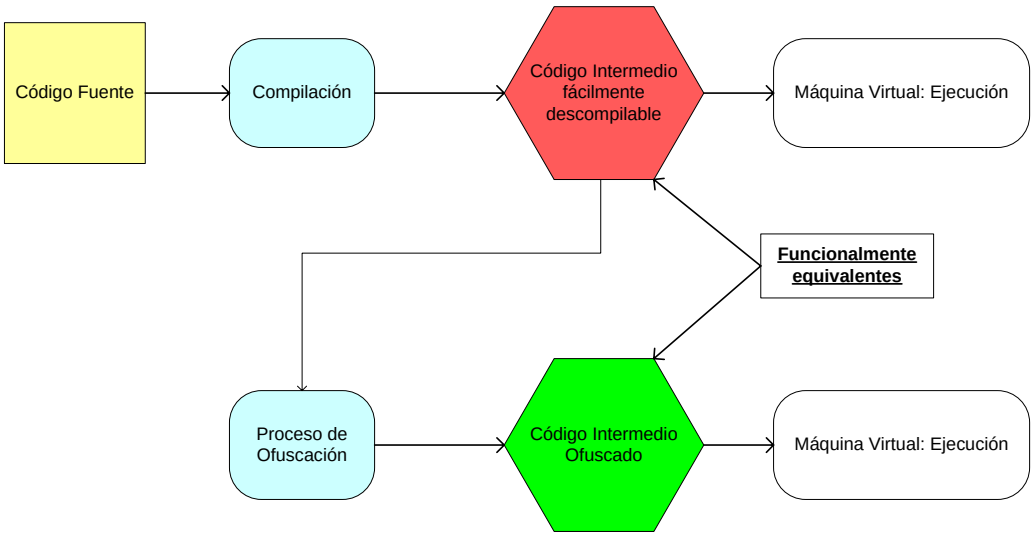


Figura 1: Ofuscación de Código Intermedio

En pocas palabras, algunas de las técnicas de ofuscación más comunes consisten en la inclusión de bucles irrelevantes, cálculos innecesarios, comprobaciones fuera de contexto, nombres de funciones y de variables que no tienen nada que ver con su cometido, funciones que no sirven para nada, interacciones inverosímiles entre variables y funciones, etc. Otras técnicas, sin embargo, son mucho más potentes, en el sentido de que requieren un conocimiento superior de las características del lenguaje, e incluso pueden estar diseñadas para burlar a herramientas de ingeniería inversa específicas.
A continuación, discutimos conceptos claves presentados por Collberg[1].

2.1 Transformación de ofuscación

Sea $P \xrightarrow{\tau} P'$ la transformación de un código fuente o intermedio P en un código fuente o intermedio P'

$P \xrightarrow{\tau} P'$ es una transformación de ofuscación si P y P' tiene el mismo comportamiento observable, entendido desde el punto de vista de lo que percibe el usuario.

Más precisamente, para que τ sea una transformación válida se debe verificar que:

- Si P falla al terminar dada una entrada, P' podría o no terminar.
- De otra forma, dada una entrada, P' debe terminar y producir el mismo resultado que P.

Idealmente, P' debería tener características que dificultan la compresión del código fuente. Se define, de manera informal, a comportamiento observable como la percepción del usuario acerca de la entrada y salida del usuario. P' podría tener comportamientos diferentes, cuyo objetivo es confundir a un posible desofuscador y/o a un hacker, que serían válidos mientras no sean parte de la experiencia del usuario de P'.

Las transformaciones de ofuscación pueden clasificarse en cuatro categorías o tipos [1]:

- Léxicas o de Estructura: renombramiento de identificadores, cambio de formatos.
- Ofuscaciones de Datos: encriptación de recursos embebidos, encriptación de metadatos, encriptación de cadenas almacenadas, modificación de jerarquías, unificación de variables.
- Ofuscaciones de Control: reconversión de flujos de control, reordenamientos de sentencias, bucles y expresiones, extensión de condiciones de loop.
- Ofuscaciones Preventivas: destinadas exclusivamente a provocar malfuncionamiento a herramientas de descompilación.

2.2 Calidad de una Transformación de Ofuscación

La calidad de una transformación de ofuscación se evalúa según los siguientes criterios: cuanto más difícil se vuelve de entender el código fuente por un lector humano (potencia), qué tan difícil resulta para una herramienta automática revertir la transformación (resistencia), qué tan bien las modificaciones introducidas se disimulan o mezclan con el resto del programa (stealth) y cuanto costo extra computacional se agrega a raíz de la aplicación de la transformación (costo).

2.2.1 Potencia

La potencia de una transformación indica, de una manera orientativa, en qué medida el código ofuscado es más difícil de comprender por un lector humano. Si bien el concepto "más difícil de comprender" no puede ser cuantificado objetivamente, se utilizan métricas de la ingeniería de software que miden la claridad conceptual y la mantenibilidad de un código como referencia.
A efectos prácticos se clasifica la potencia de cualquier transformación en baja, media y alta.

2.2.2 Resistencia

Un atacante merecedor de consideración seguramente dispondrá de herramientas configurables de ingeniería inversa, llegando al extremo de desarrollar las mismas y poder adaptarlas a las técnicas de ofuscación que vaya detectando. Debido a esto, la *resilience* o resistencia puede expresarse como la combinación de dos medidas:

Esfuerzo de Programación: la cantidad de tiempo que llevaría construir una herramienta de ingeniería inversa que efectivamente pueda reducir la potencia de una transformación τ.

Esfuerzo de Desofuscación: los recursos (tiempo / espacio) necesarios para que dicha herramienta efectivamente reduzca la potencia de una transformación τ.

Es importante entender la distinción entre potencia y resistencia. Una transformación es potente si logra confundir a un lector humano, mientras que es resistente si torna difícil la construcción de un desofuscador o hace que la ejecución del mismo se torne impráctica en entornos reales.

Las transformaciones más resistentes son aquellas que son irreversibles. Consisten generalmente en la eliminación de información presente en el programa pero que no es necesaria para la ejecución del mismo, como los nombres significativos de identificadores, entre otros.
Otras transformaciones, como el agregado de código basura que no afecta el comportamiento observable del programa, podrían ser revertidas con distintos niveles de dificultad.

2.2.3 Stealth

Es posible crear técnicas de ofuscación que modifiquen un programa de manera de hacerlo muy difícil de comprender (alta potencia) y que a su vez no sea fácil extraerlas para obtener el código original (alta resistencia).
Un ejemplo podría ser modificar la codificación de valores de variables. En vez que asignar a una variable el valor escrito en el código fuente, una técnica de ofuscación podría asignar valores enormes (del rango de los millones) y aplicar las mismas fórmulas a los valores con los que interactúa esta variable. De esta manera, comparaciones simples del tipo `while (I <= 10)` podrían transformarse en `while ((I * f(I) – 234)^12 <= 5748951478)` siendo la ejecución equivalente.
Este último código ofuscado, sin embargo, salta a la vista como sintético y forzado, y un atacante experimentado lo identificará enseguida como resultado de la aplicación de una técnica de ofuscación. Para mejorar el stealth, el código incorporado por un ofuscador debería parecerse lo más posible al código original, lo cual es un desafío dado que un código que podría ser *stealthy* en un programa podría no serlo en otro de estilo y dominio diferente.

2.2.4 Costo

La aplicación de muchas técnicas de ofuscación, como la del ejemplo de la definición de stealth, implican de manera clara un overhead en el tiempo de la ejecución del programa debido a mayores operaciones (overhead termporal). Otro tipo de ofuscaciones podrían aumentar los requerimientos de recursos espaciales (típicamente memoria) de un programa ofuscado con respecto a su versión original, mientras que algunas transformaciones no incluyen overhead, como el renombrado de identificadores. A medida que aumenta el costo de una transformación disminuye la calidad de la misma.

3 Predicados Opacos

El presente trabajo se enfoca en los predicados opacos, que son el bloque básico de las transformaciones de ofuscación que oscurecen el programa modificando el flujo de control del mismo[2].
Las transformaciones de flujo de control generalmente realizan alguna de estas tres acciones:

- Ocultar el verdadero flujo de control de un programa entre sentencias irrelevantes que no contribuyen a la ejecución del programa.
- Introducir, en el código intermedio, secuencias de control sin correspondencia en el lenguaje de alto nivel en el que originalmente fue escrito el programa.
- Remover construcciones reales del flujo de control y/o introducir construcciones falsas.

Los predicados opacos son expresiones que no pertenecen al programa a ofuscar sino que son introducidos por el ofuscador. La realidad indica que, es el predicado opaco el que realmente hace que se ejecute el código que el programador pretendía ejecutar, y no el código basura o irrelevante insertado por la herramienta de ofuscación.
Informalmente, una variable *V* es opaca si tiene alguna propiedad que es conocida a priori por el ofuscador, pero es difícil de deducir para un ingeniero inverso.
Lo mismo puede decirse para un predicado *P* cuyo valor booleano es conocido por el ofuscador, pero no por el ingeniero inverso.
Por ejemplo, una variable opaca *V* introducida por el ofuscador de valor "10" puede usarse para generar expresiones verdaderas o falsas preguntando, por ejemplo si `V == 10`, si `V < 6`, etc. El ofuscador conoce el valor de *V* en cada momento dado que es el encargado de asignarla. Sin embargo, cuanto más difícil sea para el ingeniero inverso deducir que en tal punto del programa *V* tiene valor "10", mejor funcionará el predicado opaco.
La creación de predicados opacos que sean difíciles de deducir por el ingeniero inverso es uno de los desafíos más importantes para el creador de herramientas de ofuscación. De hecho, los predicados opacos son la clave para la resistencia de las transformaciones de control.
Utilizaremos las mismas medidas que han sido definidas para transformaciones (potencia, resistencia, stealth y costo) para los predicados opacos.

3.1 Uso de Predicados Opacos

Los predicados opacos son claves en las transformaciones de ofuscación que consisten en:

- Inserción de código muerto o irrelevante. El código muerto nunca debe ejecutarse. Su existencia solo tiene por objeto confundir a un posible atacante. La no ejecución de código muerto se deja en manos de un predicado opaco. Por ejemplo, el código muerto podría situarse en el bloque `else` de un `if` cuya expresión sea un predicado opaco que evalúa siempre a verdadero.
- Extensión de Condiciones de loop. Es posible oscurecer un bucle en el programa haciendo más compleja su condición de terminación. La idea consiste en utilizar predicados opacos de valor conocido para extender la expresión que determina el bucle.
- Conversión de flujo de control de Reducible a No Reducible: los lenguajes en los cuales se basa la necesidad de la existencia de la ofuscación, como son el Java y aquellos de la plataforma .NET, son compilados o traducidos a un lenguaje intermedio, denominados bytecode y MSIL respectivamente. La característica fundamental de estos lenguajes intermedios es que son más poderosos que los lenguajes originales. Esto debe ser así, debido a que no es posible que existan construcciones en los lenguajes de alto nivel que no puedan ser transformadas a código intermedio. La idea es, construir en lenguaje intermedio un flujo de control que no tenga equivalente de alto nivel, pero preservando la ejecución correcta mediante el uso de predicados opacos. Un ejemplo sería una bifurcación condicional a una sentencia que esté en el medio de un bloque `while`, protegida mediante un predicado opaco que evalúa siempre a falso de manera que la verdadera ejecución no es alterada.

3.2 Construcción de Predicados Opacos

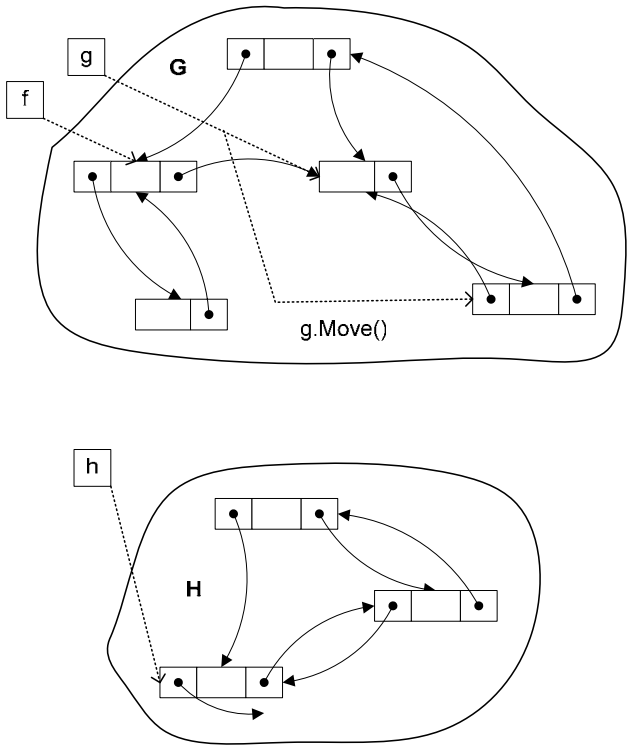
Los predicados opacos son vitales hasta el punto que la resistencia de las transformaciones está relacionada directamente con la calidad de dichos predicados
Los predicados obvios que podrían pensarse, como `P == 0`, `Q != null`, tienen resistencia a lo sumo, débil, siendo la mayoría triviales. Esto significa que un desofuscador automático podría deducir su valor mediante un análisis estático, local o global, sin insumir mucho esfuerzo. Obviamente, es necesario un nivel de protección mayor, identificando expresiones opacas cuyo esfuerzo requerido, en el peor caso, sea exponencial con respecto al tamaño del programa, pero que solo requiera un tiempo lineal o polinomial para construirlos.

3.3 Técnicas avanzadas

Existen varias técnicas avanzadas de construcción de predicados opacos. Sin embargo, tienen dos problemas fundamentales. Uno de ellos es el costo y el otro es la visibilidad del salto del flujo de control.
A continuación, describimos las técnicas más avanzadas de construcción de predicados opacos.

3.3.1 Construcción de Predicados Opacos usando Objetos y Alias

Los análisis estáticos de cualquier tipo sobre un programa se tornan significativamente costosos cuando existe la posibilidad de que existan alias para los objetos. Se ha demostrado que el análisis estático de código fuente cuando existe aliasing de objetos es NP-Hard [9] o incluso indecidible [10].
La idea básica es construir una estructura dinámica compleja con punteros y alias y mantener un conjunto de punteros a esta estructura. De esta manera, el ofuscador sabe si *p* es igual a *q* (siendo *p* y *q* punteros de una estructura) pero un desofuscador no podría saberlo realizando un análisis estático. En la figura 2, mostramos un posible ejemplo de generación de predicados opacos.




```
Node g,h;
method P(…, Node f)

/* 1 */  g= g.Move();
         h= h.Move();
/* 2 */  h= h.Insert(new Node);
         .
         .
         .
/* 3 */  x.R(…, f.Move());
         .
         .
         .
/* 4 */  if (f==g)?
/* 5 */  if (g==h)F
         .
         .
         .
/* 6 */  f.Token= False;
         g.Token= True;
         .
         .
         .
/* 8 */  f.Token= True;
         h.Token= False;
/* 9 */  if (f.Token)T
```

Figura 2: Predicados opacos construidos a partir de objetos y alias.

Se construye una estructura dinámica formada por Nodos. Cada nodo cuenta con un campo booleano Token y dos campos puntero que pueden apuntar a otros nodos. La estructura está diseñada por dos componentes conectados, G y H. Existen dos punteros globales, g y h, que apuntan a G y a H respectivamente.

3.3.2 Construcción de Predicados Opacos usando Threads

Los programas paralelos o multiproceso son mucho más difíciles de analizar estáticamente que sus contrapartidas secuenciales. La razón es el intercalamiento: si *n* segmentos de código pueden ejecutarse de manera paralela, la cantidad de formas en las que puede ejecutarse es *n!*. Si *n* es grande, rápidamente puede intuirse la dificultad de realizar un análisis estático.
La técnica es la misma que la del punto anterior, pero sumando la complejidad que brinda el multithreading.

3.4 Problemas detectados

Las técnicas analizadas arriba son las técnicas de construcción de predicados opacos más avanzadas del estado del arte. Sin embargo, en este trabajo sugerimos que no son prácticas ni viables porque implican un alto costo y no logran el stealth que sería necesario para justificarlo.

3.4.1 Costo

En ambas técnicas (Objetos-Alias y Multithreading) se observa la creación y el mantenimiento de una estructura de grafos que es totalmente ajena al programa original solo al efecto de preguntar, en algún momento, si $P == Q$ o alguna pregunta similar. Puede verse que la técnica de objetos-alias

implica un costo temporal (el tiempo de procesamiento requerido para generar y mantener la estructura) y también un costo espacial (la cantidad de memoria).
Para el caso del multithreading, que consiste en generar lo mismo mediante threads, hay que considerar también el overhead de los cambios de contexto.
Además, en ambos casos, hay una consideración práctica: los programas suelen fallar. Cuando esto ocurre, los entornos informan al usuario con un *dump* del *stack trace*, lo cual suele ser útil al desarrollador para identificar las causas de la falla. En el caso de una falla generada por un error del programa es posible que el *stack trace* esté contaminado por las llamadas del ofuscador, lo que también es un problema.

3.4.2 Visibilidad

En este trabajo sugerimos que, en realidad, no es tan importante la complejidad del cálculo del predicado opaco sino su visibilidad.
Es posible mantener una estructura enorme, compleja y cara para que el ingeniero inverso no pueda nunca llegar a saber si $P == Q$, pero eventualmente este ingeniero inverso detectará que P y Q no tienen nada que ver con el programa que está analizando y llegará a la conclusión de que **$P == Q$ es un predicado opaco,** por más que no pueda conocer su valor en cada momento que es invocado.
El hecho de identificar predicados opacos ya es una información invaluable a los efectos de desofuscar el programa, dado que el ingeniero sabe que algunos de los dos casos (verdadero o falso) esconde código resultado de transformaciones, y ese conocimiento podría ser suficiente para lograr los propósitos de la ingeniería inversa.
Una herramienta inicial para detectar predicados opacos es, sencillamente, identificar a todas las sentencias de salto condicional del código intermedio, sabiendo que algunas corresponden a la lógica del programa y que otras son introducidas por el ofuscador.
Esto es casi trivial. De hecho, la herramienta ildasm.exe provista por Microsoft que extrae el código intermedio a partir de ejecutables portables .NET ya marca con un espacio aquellas instrucciones que son un salto, condicional o no, haciendo muy fácil identificar los predicados opacos potenciales.

4 Predicados Opacos Superiores: utilización de bloques try-catch-finally

En esta sección, introducimos nuestra propuesta para generar predicados opacos de mayor calidad.

4.1 Bloques Try-Catch-Finally

Los lenguajes sobre los que trata este trabajo, implementan el manejo de excepciones basado en bloques *try-catch-finally*. No es el objetivo de este trabajo describir en detalle esta construcción. Basta comentar que *try, catch y finally* son palabras clave que delimitan bloques disjuntos de código de manera que, si ocurre una excepción, de cualquier tipo, dentro del bloque *try* (la excepción puede ocurrir incluso en otro método de otra clase, o incluso en otro módulo que puede estar en un archivo binario distinto) la ejecución se deriva, automáticamente, a la primera sentencia del bloque *catch*. Si no ocurre ninguna excepción, el bloque *catch* no se ejecuta.
El bloque *finally*, que es optativo, se ejecuta en todos los casos y suele utilizarse para realizar tareas de limpieza que deben realizarse tanto si ocurrió una excepción, como si no. El ejemplo clásico de un bloque *finally* consiste en el cierre de una conexión a una base de datos.

4.2 Predicados Opacos utilizando Excepciones

La propuesta de este trabajo consiste en la utilización de predicados opacos simples y de relativamente poco costo (como $q == 0$, $p == null$) pero prescindiendo del uso de una sentencia de salto condicional para implementarlo y, en cambio, forzando una excepción.
De esta manera, si bien los valores son simples, a un ingeniero inverso le resultaría difícil identificar los posibles lugares donde efectivamente ocurre el salto condicional mediante un análisis estático.
A continuación mostramos ejemplos utilizando el lenguaje de código intermedio de .NET, llamado MSIL. Está fuera del alcance de este trabajo explicar el funcionamiento del MSIL, remitiendo a las especificaciones de la ECMA al respecto [7, 8].

Ejemplo: Predicado Opaco utilizando una sentencia IF. Corresponde a un if (Predicado Opaco False) then (código real) else (código bogus)

```
IL_0000:  ldarg.1
IL_0001:  brtrue.s   IL_003e          /*** salto condicional ***/
      ..........  /* Código real de la aplicación */
IL_0033:  ldstr      "Codigo real"
IL_0038:  call       void [mscorlib]System.Console::Write(string)
IL_003d:  ret
      ..........  /* Código falso */
IL_003e:  ldstr      "Codigo Bogus introducido por el ofuscador"
IL_0043:  call       void [mscorlib]System.Console::Write(string)
IL_0048:  ret
```

Nótese la línea IL_0001 (resaltada), con la sentencia de salto condicional *brtrue* sobre el argumento 1 del método (ldarg.1). Es importante destacar que, por muy complicada que sea la manera en la que el ofuscador oculta que ese parámetro es cero, es suficiente con saber que el predicado opaco está allí, quedando solamente analizar los bloques *then* y *else* y el comportamiento de los mismos.

Ejemplo: Predicado Opaco Utilizando Bloques *Try-Catch*

```
IL_0000:  ldc.i4.0
IL_0001:  stloc.0
.try
{
      ..........  /* Código real */

  IL_0032:  ldstr      "Codigo real"
  IL_0037:  call       void [mscorlib]System.Console::Write(string)
  IL_003c:  ldloc.0
  IL_003d:  ldarg.1
  IL_003e:  div
      ..........  /* Código falso */
  IL_003f:  call       string [mscorlib]System.Convert::ToString(int32)
  IL_0044:  call       void [mscorlib]System.Console::Write(string)
  IL_0049:  ldstr      "Codigo Bogus introducido por el ofuscador"
  IL_004e:  call       void [mscorlib]System.Console::Write(string)
  IL_0043:  leave.s    IL_0032
}  // end .try
catch [mscorlib]System.Object
{                  ..........  /* Código real */

  IL_0055:  pop
  IL_0056:  ldstr      "Codigo real"
  IL_005b:  call       void [mscorlib]System.Console::Write(string)
  IL_0050:  leave.s    IL_0032
}  // end handler
IL_0052:  ret
```

Un ejercicio interesante podría ser intentar identificar en que instrucción se encuentra el predicado opaco sabiendo que el valor del parámetro es, al igual que en el caso anterior, un entero de valor cero.
La respuesta es, en la línea *IL_003e: div*. Lo que esta sentencia hace es provocar división por cero. El resultado neto es, entonces, la continuación de la ejecución en el bloque *catch* donde reside el resto del código real.
Lo importante de este esquema es que el error inducido puede estar en cualquier parte dentro de un bloque de código, sin tener como requerimiento el uso de una sentencia de salto condicional que son fácilmente identificables. Esto convierte al predicado opaco en una sentencia más, no identificable estáticamente.
Algunas de las excepciones que podrían inducirse son división por cero, uso inválido de *null*, error de conversión de tipos, valores fuera de rango, *cast* inválidos, entre otros.

4.3 Predicados Opacos utilizando Excepciones con Stealth mejorado

Intentemos llevar este concepto aún más allá. Dado que la bifurcación del flujo de control que puede manipular el ofuscador surge de la generación inducida y controlada de un error en *runtime*, un buen ofuscador podría utilizar construcciones comunes del programa siendo ofuscado e inducir errores gracias a los valores opacos conocidos, pero generando estructuras exactamente iguales a la que utilizó el programador de la aplicación.

4.3.1 Ejemplo: uso inválido de null

Supongamos que el ofuscador detecta que el programa a ofuscar hace uso intensivo de objetos que son instancias de una clase que llamaremos *CACIC* y un método llamado *Compartir()*. Esto quiere decir que, las sentencias en IL del tipo *A.Compartir*, siendo *A* una variable instancia de la clase *CACIC* son comunes y frecuentes.
El ofuscador podría entonces ingresar predicados opacos de la forma *A.Compartir()*, pero en un momento en el cual conoce que *A* tiene valor nulo.
La ofuscación del flujo de control se realiza perfectamente: para el ingeniero inverso es todo un bloque de código coherente con construcciones normales y muy utilizadas, como por ejemplo *A.Compartir()*.
Sin embargo, desde el momento que *A* es un valor nulo, la ejecución se bifurca hacia el bloque *catch*, lo cual es muy difícil de determinar automáticamente con un análisis estático, y es muy difícil de detectar con un análisis visual del código, dado que el salto se produce en una sentencia totalmente común al programa. En consecuencia, éste predicado goza de altos niveles de stealth.

4.3.2 Ejemplo Dos: colaboración con el programador.

Si la aplicación utilizara de manera intensiva acceso a datos, con una pequeña colaboración del desarrollador, la excepción podría ser generada mediante llamados SQL erróneos. Esto confundiría aún más a un ingeniero inverso desprevenido, ya que es, a priori, impensable que un mecanismo automático como un ofuscador introduzca sentencias de llamado a datos.

4.4 Algunas limitaciones

Este esquema presenta algunas limitaciones. Una de ellas consiste en que los bloques *try-catch-finally* pueden anidarse pero nunca solaparse, de manera que no es posible realizar cualquier construcción arbitraria. Otra limitación es, claramente, que es necesario respetar los bloques *try-catch-finally* que son verdaderamente parte del programa, respetando el funcionamiento del mismo.

5 Conclusiones

La ofuscación es la técnica estándar para la protección del código fuente en ambientes de desarrollo modernos. Dentro de las técnicas de ofuscación, los predicados opacos son el bloque de construcción básico y su fortaleza determina en gran medida la calidad de las transformaciones de ofuscación aplicadas.
Hemos analizado las técnicas de generación de predicados opacos más avanzadas, como la utilización de estructuras utilizando alias y multithreading y concluímos que por su elevado costo, tanto temporal como espacial, no son la solución óptima.
La propuesta de este trabajo ha sido la creación de predicados opacos utilizando el mecanismo de manejo de excepciones como una manera de generar predicados de mayor calidad mediante la reducción del costo y el aumento del stealth.
Como líneas de trabajo futuro podemos mencionar el estudio en mayor detalle del uso de características avanzadas de los ambientes modernos para la protección de la propiedad intelectual, como podrían ser el uso de atributos personalizados y el mencionado manejo de excepciones. También hay un gran campo por explorar en el estudio y análisis del concepto de "funcionalidad equivalente" entre un programa sin ofuscar y un programa ofuscado.

REFERENCIAS

[1] Christian S Collberg, Clark Thompson. Watermarking, Tamper-Proofing, and Obfuscation - Tools for Software Protection,

[2] Christian S Collberg, Clark Thompson, Douglas Low. Manufactoring Cheap, Resilient, and Stealthy Opaque Construct

[3] Willy Alexánder Marroquín. Ofuscadores (De la protección relativa del código intermedio), http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art146.asp

[4] Mr. Jeff Hughes, Dr. Martin R. Stytz, Ph.D. Advancing Software Security– The Software Protection Initiative, 2001

[5] Microsoft Software Developer Network, Microsoft Visual Studio 2005

Evaluation Guide. http://msdn.microsoft.com/vstudio/tour/evaluation/default.htm

[6] Revista Information Technology. Suplemento Especial Seguridad, Junio 2005.

[7] The Common Language Infrastructure (CLI) Partition II*: Metadata Definition and Semantics* http://msdn.microsoft.com/net/ecma/; http://www.ecma-international.org/publications/standards/Ecma-335.htm.

[8] The Common Language Infrastructure (CLI) Partition III: *CIL Instruction Set* http://msdn.microsoft.com/net/ecma/; http://www.ecma-international.org/publications/standards/Ecma-335.htm.

[9] S. Horwitz. Precise flow insensitive May-Alias analysis is NP-Hard. TOPLAS, 19(1):1-6, 1997.

[10] G. Ramalingam. The undecidability of aliasing. TOPLAS, 16(5):1467-1471, Septiembre 1997.

[11] Fortune 500. http://en.wikipedia.org/wiki/Fortune_500

[12] D. Dolz, G. Parra. Ofuscadores de Código Intermedio. Reporte Preliminar. *WICC* 2006.